

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 27.01.2024 12:02:46
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

Утверждаю
Проректор по учебной работе
О.Г. Локтионова
«25» 09 2023 г.



ИНТЕЛЛЕКТУАЛЬНАЯ ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ В БИОТЕХНИЧЕСКИХ СИСТЕМАХ

Методические рекомендации по выполнению лабораторных работ
для студентов направления подготовки 12.04.04 – «Биотехнические
системы и технологии» (магистр)

Курск 2023

УДК 621.(076.1)

Составители: А.А.Кузьмин

Рецензент:

Кандидат технических наук, доцент *Т.Н. Конаныхина*

Интеллектуальная поддержка принятия решений в биотехнических системах: методические рекомендации по выполнению лабораторных работ для студентов направления подготовки 12.04.04 – «Биотехнические системы и технологии» (магистр) / Юго-Зап. гос. ун-т; сост.: А.А.Кузьмин. - Курск, 2023. - 115 с.

Содержат методические рекомендации к проведению лабораторных работ по дисциплине «Интеллектуальная поддержка принятия решений в биотехнических системах». Методические указания по структуре, содержанию и стилю изложения материала соответствуют методическим и научным требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для студентов направления подготовки 12.04.04 – «Биотехнические системы и технологии» (магистр)

Текст печатается в авторской редакции

Подписано в печать 25.09.23 Формат 60x84 1/16

Усо.печ.л. 6,7. Уч.-изд.л. 6,1. Тираж 30 экз. Заказ: 1087. Бесплатно.

Юго-Западный государственный университет.

305040. г. Курск, ул. 50 лет Октября, 94.

ЛАБОРАТОРНАЯ РАБОТА №1. ВЫДЕЛЕНИЕ ИНФОРМАТИВНЫХ ПРИЗНАКОВ.

Цель работы: овладение навыками формирования множества информативных признаков, характеризующих состояния биообъектами, статистическими методами.

Краткие теоретические сведения

В процессе обработки результатов исследования поведения биообъекта с целью построения решающих диагностических правил или иных исследований возникает проблема разрешения противоречия между многомерностью исходного признакового пространства и стремлением упростить модель для выделения наиболее существенных связей. Накопленный опыт информационного анализа свидетельствует о том, что любые статистические операции становятся более эффективными при уменьшении числа используемых признаков. Снижение размерности исходного пространства состояний осуществляется двумя основными путями:

- отбор признаков удовлетворяющих определенному критерию или целевой функции;
- заменой групп признаков на определенный функционал от данной группы.

Наиболее информативный признак обладает следующими свойствами:

- 1) он наиболее представителен в классе (свойство удовлетворяющей исследователя диагностической специфичности и чувствительности);
- 2) он наименее линейно зависим с другими (свойство ортогональности признакового пространства);
- 3) он достаточно вариативен как внутри класса, так и вне его (свойство репрезентативности).

Большинство алгоритмов формирования множества информативных признаков путем применения алгоритмов селекции работают в двух режимах: с заданием количества оставленных признаков (режим А) и с автоматическим отбором без априорно заданного количества (режим Б). Все алгоритмы селекции могут работать как с первичным признаковым пространством, так и с уже обработанным (т.е. после одного и более режимов селекции).

К наиболее простым статистическим методам селекции признаков относятся:

- По относительному дисперсионному разбросу (вариации);

- По парной корреляции;
- По решаемым спорам;
- По качественной вариации;
- По информационному показателю силы влияния (ИПВ);
- По критериям различий выборок в различных классах (анализ доверительных интервалов, коэффициент Стьюдента и т.п.);
- По информационным критериям (Кульбака, анализ энтропийных характеристик и т.п.);
- По комплексным критериям, представляющие собой аддитивную, мультипликативную, дробно-рациональную, логическую или иные формы интеграции критериев различной семантики.

Перед проведением селекции по указанным критериям необходимо провести процедуры удаления артефактов и восстановления пропущенных значений.

Например, для восстановления пропущенных значений в признаках в случае их численного представления рекомендуется использовать метод линейной экстраполяции по двум ближайшим соседям, выбор которых осуществляется с учетом принципа несмещенности ошибки.

Математический аппарат механизмов селекции заключается в следующем:

- По относительному дисперсионному разбросу (вариации).

Каждый признак характеризуется величиной $V = 100 * D / E$, где D , E , V - соответственно, оценки среднеквадратичного отклонения, математического ожидания и вариации признака.

В режиме «А» пользователь задает N_{sel} - количество признаков, которое необходимо оставить после процедуры отбора.

После определения V_i , $i=1, koolpr$, строится гистограмма, по которой, начиная с верхней границей, подсчитывается количество признаков. При достижении границы N_{sel} осуществляется при необходимости процедура удаления признаков, не отвечающих требованию селекции, с использованием датчика случайных чисел равномерного закона распределения. В режиме «Б» определяется пороговое значение показателя V ; оставляются те признаки, у которых найденное V_i превышает указанное значение.

-По парной корреляции.

В связи с большой размерностью исходного признакового пространства в данном случае предлагается отойти от принципа построения полной матрицы корреляционной связи, осуществление которого приведет к

возникновению проблемы «проклятия многомерности». Предлагается следующий механизм селекции.

Случайным образом, используя ДСЧ с равномерным законом распределения, строятся вектора:

$$R[k,i], k=1,2,3 \quad i=1, \text{коолпр},$$

где $R[k,i]$ - ранжированное с точностью до 0.1 абсолютное значение коэффициента парной корреляционной связи между признаками k и i .

Селекция по желанию пользователя осуществляется в 3 этапа:

- выбираются те факторы, у которых $R[1,i] \& R[2,i] \& R[3,i] \leq R_{\text{пор}}$;
- выбираются те факторы, у которых по i два из трех $\leq R_{\text{пор}}$;
- выбираются те факторы, у которых по i хотя бы один из трех $\leq R_{\text{пор}}$.

В режиме А (задано N_{sel}) последовательно повторяются три этапа с определенным пороговым значением коэффициента парной корреляции (рекомендуется $R_{\text{пор}}=0.14$), а затем, с использованием ДСЧ, селективируются «лишние» признаки.

-По решаемым спорам.

Концепция этого метода заключается в том, что отбираются те признаки, которые имеют наименьшее количество «споров» - перекрытий доверительных интервалов по классам состояний биообъекта. Алгоритм в данном случае заключается в следующем.

Для каждого признака определяется доверительный интервал с определенным уровнем статистической значимости: $M_k \mp T_k$, где M_k -оценка модального значения признака, T_k – «трубка допуска» в k -ом классе.

Затем определяются величины G_{min} и G_{max} - минимальная и максимальная границы доверительных интервалов в различных анализируемых классах. Весь диапазон $[G_{\text{min}}, G_{\text{max}}]$ разбивается на равностоящие диапазоны, на которые «накладываются» интервалы признака из разных классов: если в диапазоне нет пересечения, то он приобретает значение «номера класса», в противном случае - номера «очередного класса». Т.о., определяется количество «новых» классов и диапазоны «перекрытия».

Если задано количество селективируемых признаков (режим «А»), то для

каждого признака определяется значение критерия: $K_r = \frac{N_{kn} \cdot \sum_{i=1}^{N_k} \frac{\max(l, ll)}{\min(l, ll)}}{N_k}$, где

N_{kn} - количество "новых" классов, N_k - первоначальное количество классов, l, ll -соответственно «длины» двух соседних классов в зонах перекрытия.

Суммирование ведется, если $\min(l, ll) > 0$ и $\max(l, ll)$ не больше N_k .

Затем отбирается N_{sel} признаков с минимумом K_r .

В режиме «Б» пользователь задает максимально допустимую величину перекрытий и отбираются те признаки, у которых полученное количество перекрытий не превышает порога.

-По комплексному критерию (частный случай).

Данный метод аналогичен во многом описанному ранее механизму селекции по корреляции, но используется одновременно два критерия: $V_{\text{порог}}$ и $R_{\text{порог}}$ (количественная вариация и корреляция).

Для каждого признака определяется значение критерия: $K_i = V_i * \left(\frac{1}{R[1,i]} + \frac{1}{R[2,i]} + \frac{1}{R[3,i]} \right)$, где $R[1,i]$, $R[2,i]$, $R[3,i]$ - три наименее коррелированных с признаком i значений коэффициентов корреляции.

В режиме «Б» задаются значения $V_{\text{порог}}$ и $R_{\text{порог}}$; оставляются те признаки, у которых $K_i \geq 3 \cdot V_{\text{порог}} / R_{\text{порог}}$.

В режиме «А» оставляется заданное количество признаков с максимальным значением K_i согласно построенной гистограмме этих значений и применения процедуры случайного выбора.

-По качественной вариации.

Данный вид отбора подобен селекции по количественной вариации, т.е. в качестве информативных отбираются обладающие максимальным значением коэффициента качественной вариации, но не более N_{sel} признаков. Коэффициент качественной вариации оценивается по формуле:

$Q_i = \frac{l \cdot (N^2 - B)}{(l-1) \cdot N^2}$, где N - количество объектов, l - количество классов-диапазонов, B - сумма квадратов числа попаданий значений признаков в классы-диапазоны.

Примечание: Применять данный критерий следует осторожно, т.к. он не сколько отражает свойства признака в смысле деления на классы, сколько говорит о равномерности или неравномерности распределения значений признака по классам-диапазоном. Т.е. в случае большой зашумленности наилучшими будут признаки с наиболее приближенным распределением частот своих значений к равномерному закону распределения.

-По информационному показателю силы влияния (ИПВ).

Данный критерий отсева основывается на свойствах информационной энтропии и является, пожалуй, наиболее эффективным, хотя и наименее быстрым из представленных ранее. Для каждого признака i строится гистограмма его значений, по которой оценивается общая энтропия E_i . Поступая аналогичным образом в каждом классе, получаем внутриклассовые энтропии E_k . Тогда негэнтропия равна:

$\hat{Y}_i = \frac{Y_i - (n_1 * Y_{k1} + \dots + n_k * Y_{kn})}{N}$, где n_j - количество объектов в классе j , N - общее к-во объектов.

Отсюда имеем критерий: $\hat{A}_i = \sqrt{\frac{\hat{Y}_i}{Y_i}}$.

Селектируются N_{sel} признаков с максимальными значениями ИПВ.

Порядок выполнения лабораторной работы

- 1) Изучить теоретический материал;
- 2) Произвести «ручной» выбор информативных признаков (по таблицам 1,2 Приложения) различными способами и сравните результаты;
- 3) Осуществить выбор информативных признаков с помощью стандартного инструментария электронных таблиц или интеллектуальных калькуляторов.
- 4) По результатам п.2 и п.3 оформите отчет.

Контрольные вопросы

1. Какие диагностические критерии используются при селекции наиболее информативных признаков?
2. Какие статистические критерии используются при селекции наиболее информативных признаков?
3. Какие интеллектуализированные программные средства используются для формирования информативного признакового пространства?
4. Каким образом влияет изменение мерности признакового пространства на качество диагностики?
5. Каким образом осуществляется селекция артефактов результатов мониторинга?
6. Каким образом восстанавливаются пропущенные данные?
7. Как использовать доверительные интервалы для оценки информативности признаков?
8. Как использовать доверительные интервалы для оценки риска их использования в качестве инструмента формирования множества репрезентативных признаков?

Приложение

Таблица 1 Загрязнения города 1 (фрагмент мониторинга, у.е.)

годы	Пыль	Оксид уг- лерода	Диоксид азо- та	фенол	формальдегид	марганец
1	8	4,5	6,37	11	4,20	1,76
2	6	1,7	4,68	11	3,51	0,40
3	8	4,3	1,69	5	3,81	2,33
4	4	3,1	7,84	5	3,51	2,33
5	6	5,4	0,77	5	3,47	1,75
6	6	5,4	0,77	2	3,47	2,15
7	8	5,0	1,69	5	4,89	2,17
8	8	5,0	1,69	8	3,28	1,59
9	10	5,8	1,27	11	3,51	1,40
10	6	5,0	0,77	10	3,82	2,23
11	6	5,0	1,69	11	3,28	1,97
12	4	5,0	1,68	11	4,89	2,34
13	4	5,0	3,03	8	4,20	1,61
14	4	5,0	1,69	5	3,47	2,34
15	4	5,0	4,68	2	3,28	0,50
16	4	5,0	1,69	2	3,28	1,40
17	6	5,0	1,69	5	3,70	0,97
18	3	6,4	1,69	11	3,81	1,19
19	2	6,5	1,69	2	3,28	1,40
20	2	6,6	1,60	2	3,47	0,25
21	4	6,5	3,03	5	4,89	0,74

Таблица 2 Загрязнения города 2 (фрагмент мониторинга, у.е.)

годы	Пыль	Оксид уг- лерода	Диоксид азота	фенол	формальдеги д	марганец
1	6	1,5	2,667	8	3,5	0,8
2	4	0,667	2,222	8	2,75	0,16
3	6	1,45	1,333	6	2,25	1,7
4	2	1,067	4,444	6	2,75	1,7
5	4	1,833	0,889	6	3,25	2,5
6	4	1,833	0,889	4	3,25	1,1
7	6	1,667	1,333	6	3,75	2,0
8	6	1,667	1,333	10	3,0	0,7
9	8	2	1,156	8	2,75	0,6
10	4	1,667	0,889	7,4	2,0	1,2
11	4	1,667	1,333	8	3,0	3,5
12	2	1,667	1,33	8	3,75	1,4
13	2	1,667	1,778	10	3,5	3,2
14	2	1,667	1,333	6	3,25	1,4
15	2	1,663	2,222	4	3,0	0,2
16	2	1,667	1,333	4	3,0	0,6
17	4	1,667	1,333	2	2,5	0,4
18	5	2,333	1,333	8	2,25	0,5
19	1,8	2,5	1,333	4	3,0	0,6
20	1,8	2,67	1,3	4	3,25	0,1
21	2	2,5	1,778	6	3,75	0,3

ЛАБОРАТОРНАЯ РАБОТА №2: ИСПОЛЬЗОВАНИЕ НЕЙРОННОЙ СЕТИ В ДИАГНОСТИЧЕСКОМ ПРОЦЕССЕ

Цель работы: овладение навыками применения искусственных нейронных сетей в диагностическом процессе на примере персептрона и сетей Кохонена.

Краткие теоретические сведения.

Нейронные сети представляют собой новую и весьма перспективную вычислительную технологию, дающую новые подходы к исследованию динамических и классификационных задач в системах поддержки принятия решений в диагностическом процессе и-или при обработке результатов мониторинга функционирования биообъекта (включая экологические аспекты).

Способность к моделированию нелинейных процессов, работе с зашумленными данными и адаптивность дают возможности применять нейронные сети для решения широкого класса задач. Приложения нейронных сетей охватывают самые разнообразные области интересов: распознавание образов, обработка зашумленные данных, дополнение образов, ассоциативный поиск, классификация, оптимизация, прогноз, диагностика, обработка сигналов, абстрагирование, управление процессами, сегментация данных, сжатие информации, сложные отображения, моделирование сложных процессов, машинное зрение, распознавание речи.

Первый интерес к нейросетям был обусловлен пионерской работой МакКаллока и Питса, изданной в 1943 году, где предлагалась схема компьютера, основанного на аналогии с работой человеческого мозга. Они создали упрощенную модель нервной клетки – *нейрон*. Мозг человека состоит из белого и серого веществ: белое – это тела нейронов, а серое – это соединительная ткань между нейронами, или *аксоны и дендриты*. Мозг состоит примерно из 10^{11} нейронов, связанных между собой. Каждый нейрон получает информацию через свои дендриты, а передает ее дальше только через единственных аксон, разветвляющийся на конце на тысячи *синапсов*. Простейший нейрон может иметь до 10000 дендритов, принимающих сигналы от других клеток. Таким образом, мозг содержит не менее 10^{15} взаимосвязей между нейронами, обеспечивающих его целевое востребуемое функционирование.

Искусственным нейроном называется простой элемент, сначала вычисляющий взвешенную сумму V входных величин x_i :

$V = \sum_{i=1}^N W_i * x_i = W * \bar{X}$, где N – размерность пространства входных сигналов.

Затем полученная сумма сравнивается с пороговой величиной W_0 , вслед за чем вступает в действие нелинейная функция активации f . Коэффициенты $\{W_i\}$ во взвешенной сумме обычно называют *синаптическими коэффициентами* или *весами*. Саму же взвешенную сумму V мы будем называть потенциалом нейрона i . Выходной сигнал тогда имеет вид $f(V)$.

Величину порогового барьера можно рассматривать как еще один весовой коэффициент при постоянном входном сигнале. В этом случае говорится о *расширенном входном пространстве*: нейрон с N -мерным входом имеет $N+1$ весовой коэффициент. Если ввести в уравнение пороговую величину W_0 , то оно переписывается так:

$$V = \sum_{i=1}^N W * x + W_0$$

В зависимости от способа преобразования сигнала и характера активации возникают различные виды нейронных структур. Существуют *детерминированные нейроны*, когда активизирующая функция однозначно вычисляет выход по входу, и *вероятностные нейроны*, состояние которых в момент t есть случайная функция потенциала и состояния в момент $t-1$.

В ИСКУССТВЕННЫХ НЕЙРОНАХ МОГУТ БЫТЬ РАЗЛИЧНЫЕ ФУНКЦИИ АКТИВАЦИИ, К НАИБОЛЕЕ ТИПОВЫМ ИЗ КОТОРЫХ ОТНОСЯТСЯ СЛЕДУЮЩИЕ ВИДЫ ФУНКЦИЙ:

- ◆ Линейная: выходной сигнал нейрона равен его потенциалу,
- ◆ пороговая: нейрон выбирает решение из двух вариантов: активен / неактивен,
- ◆ Многопороговая: выходной сигнал может принимать одно из q значений, определяемых $(q-1)$ порогом внутри предельных значений.
- ◆ Сигмоидная: рассматриваются два вида сигмоидных функций:

$$s = f(V) = \frac{1}{1 + \exp(-bV)}$$

с выходными значениями в промежутке $[0,1]$ и

$$s = f(V) = \frac{\exp(bV) - 1}{\exp(bV) + 1}$$

с выходными значениями в промежутке $[-1,1]$.

Коэффициент b определяет *крутизну* сигмоида. Поскольку сигмоидная функция является гладким отображением $(-\infty, \infty)$ на $(-1,1)$, то кру-

тизну можно учесть через величины весов и порогов, и без ограничения общности можно полагать ее равной единице.

Графические изображения простейшего нейрона и виды функций с их графиками приведены на рис. 1.

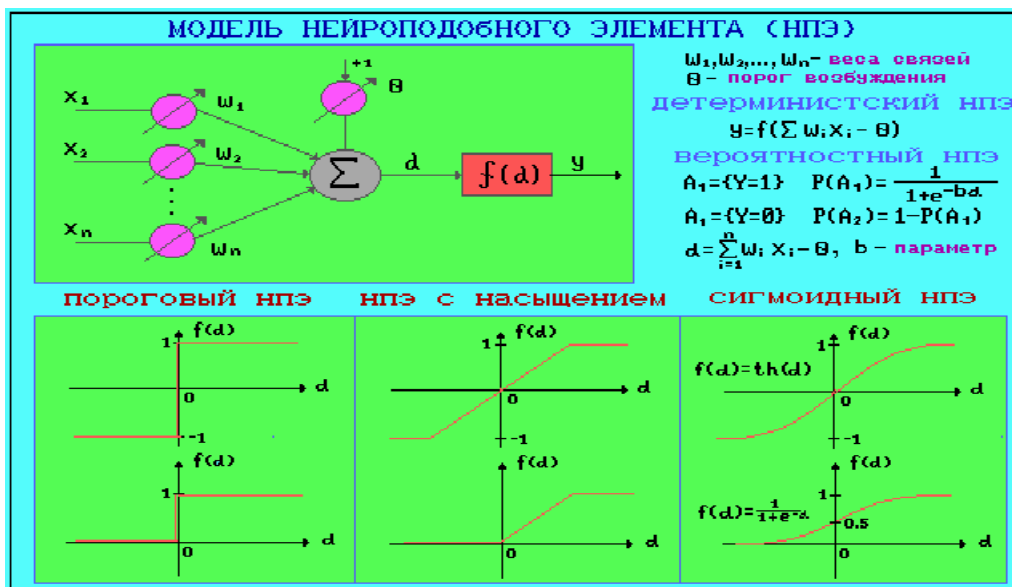


Рис.1 Изображение искусственного нейрона

Из точек на плоскости и соединений между ними можно построить множество графических фигур, называемых *графами*. Если каждую точку представить себе как один нейрон, а соединения между точками – как дендриты и синапсы, то получим нейронную сеть. Но не всякое соединение нейронов будет работоспособно или вообще целесообразно.

По архитектуре связей нейросети могут быть сгруппированы в два класса: сети прямого распространения, в которых связи не имеют петель, и сети рекуррентного типа, в которых возможны обратные связи (см. рис. 2).

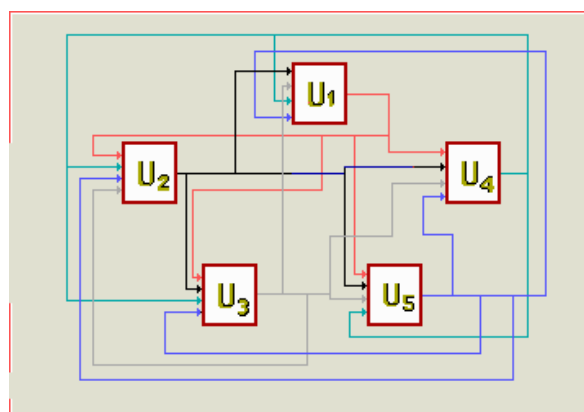
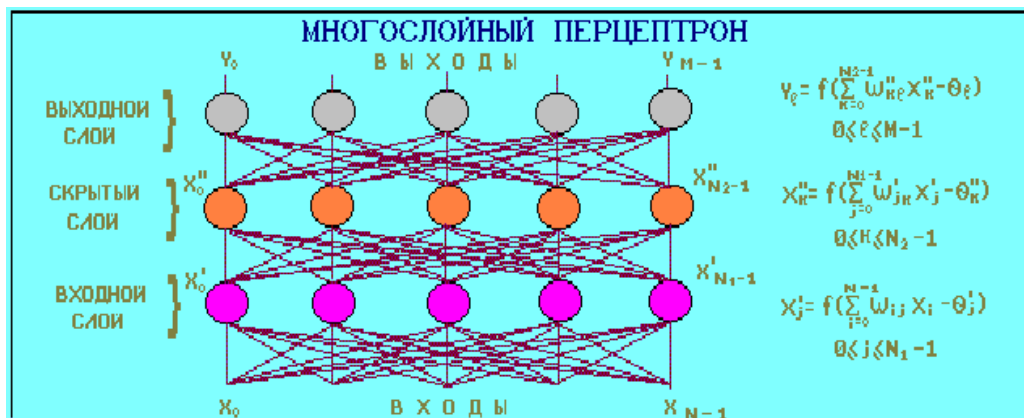


Рис. 2 Рекуррентная сеть

Сети прямого распространения подразделяются на однослойные перцептроны (сети) и многослойные перцептроны (сети). Название перцептрона для нейросетей придумал американский нейрофизиолог Ф. Розенблатт, придумавший в 1957 году первый нейропроцессорный элемент (НПЭ), то есть нейросеть. Он же доказал сходимость области решений для перцептрона при его обучении. Сразу после этого началось бурное исследование в этой области и был создан самый первый нейрокомпьютер Mark I. Многослойные сети отличаются тем, что между входными и выходными данными располагаются несколько так называемых скрытых слоев нейронов, добавляющих больше нелинейных связей в модель.

Рассмотрим устройство простейшей многослойной нейросети (схема представлена на рис.3). Любая нейронная сеть состоит из **входного слоя** и **выходного слоя**. Соответственно подаются независимые и зависимые переменные. Входные данные преобразуются нейронами сети и сравниваются с выходом. Если отклонение больше заданного, то специальным образом изменяются веса связей нейронов между собой и пороговые значения нейронов. Снова происходит процесс вычислений выходного значения и его сравнение с эталоном. Если отклонения мень-



ше заданной погрешности, то процесс обучения прекращается.

Рис. 3 Схема многослойного нейрона

Помимо входного и выходного слоев в многослойной сети существуют так называемые **скрытые слои**. Они представляют собой нейроны, которые не имеют непосредственных входов исходных данных, а связаны только с выходами входного слоя и с входом выходного слоя. Таким образом, скрытые слои дополнительно преобразуют информацию и добавляют нелинейности в модели.

Рассмотрим более подробно функционирование многослойного перцептрона, представленного на рис. 3.

Если однослойная нейросеть очень хорошо справляется с задачами классификации, так как выходной слой нейронов сравнивает полученные от предыдущего слоя значения с порогом и выдает значение либо ноль, то есть меньше порогового значения, либо единицу - больше порогового (для случая пороговой внутренней функции нейрона), и не способен решать большинство практических задач (что было доказано Минским и Пейпертом), то многослойный перцептрон с сигмоидными решающими функциями **способен аппроксимировать любую функциональную зависимость** (это было доказано в виде теоремы). Но при этом не известно ни нужное число слоев, ни нужное количество скрытых нейронов, ни необходимое для обучения сети время. Эти проблемы до сих пор стоят перед исследователями и разработчиками нейросетей.

Поведение рекуррентных сетей описывается дифференциальными или разностными уравнениями, как правило, первого порядка. Это расширяет области применения нейросетей и способы их обучения. Сеть организована так, что каждый нейрон получает входную информацию от других нейронов, возможно, и от самого себя, и от окружающей среды. Этот тип сетей имеет важное значение, так как с их помощью можно моделировать нелинейные динамические системы.

Среди рекуррентных сетей можно выделить **сети Хопфилда и сети Кохонена**.

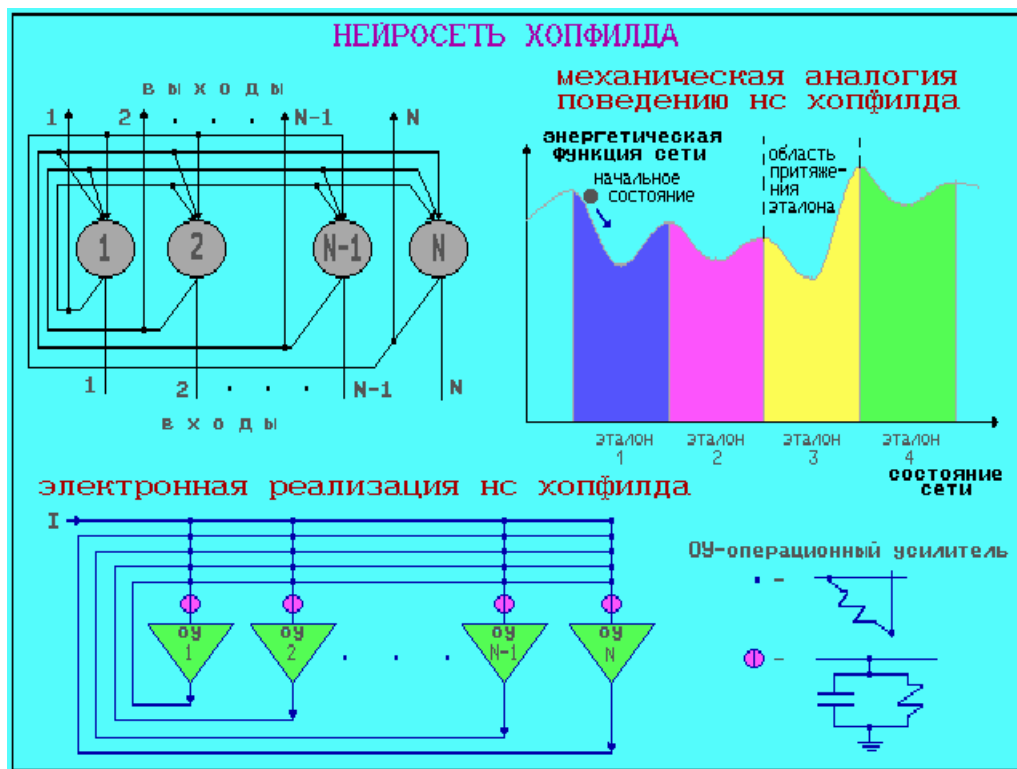


Рис.4 Структура нейросети Хопфилда

С помощью сетей Хопфилда можно обрабатывать неупорядоченные (рукописные буквы), упорядоченные во времени (временные ряды) или пространстве (графики) образцы. Рекуррентная нейросеть простейшего вида была введена Хопфилдом и построена она из N нейронов, связанных каждый с каждым кроме самого себя, причем все нейроны являются выходными. Нейросеть Хопфилда можно использовать в качестве **ассоциативной памяти**. Архитектура сети Хопфилда изображена на рис. 4. Сеть Кохонена еще называют «самоорганизующейся картой признаков». Сеть такого типа рассчитана на самостоятельное обучение во время обучения сообщать ей правильные ответы необязательно. В процессе обучения на вход сети подаются различные образцы. Сеть улавливает особенности их структуры и разделяет образцы на кластеры, а уже обученная сеть относит каждый вновь поступающий пример к одному из кластеров, руководствуясь некоторым критерием «близости». Сеть состоит из одного входного и одного выходного слоя. Количество элементов в выходном слое непосредственно определяет, сколько различных кластеров сеть сможет распознать. Каждый из выходных элементов получает на вход весь входной вектор. Как и во всякой нейронной сети, каждой связи приписан некоторый синаптический вес. В большинстве случаев каждый выходной элемент соединен также со своими соседями. Эти внутрислойные связи играют важную роль в процессе обучения, так как кор-

ректировка весов происходит только в окрестности того элемента, который наилучшим образом откликается на очередной вход. Выходные элементы соревнуются между собой за право вступить в действие и «получить урок». Выигрывает тот из них, чей вектор весов окажется ближе всех к входному вектору.

Главное отличие и преимущество нейросетей перед классическими средствами прогнозирования и классификации заключается в их способности к обучению. На этапе обучения происходит вычисление синаптических коэффициентов в процессе решения нейронной сетью задач, в которых нужный ответ определяется не по правилам, а с помощью примеров, сгруппированных в обучающие множества. Таким образом, нейросеть на этапе обучения сама выполняет роль *эксперта* в процессе подготовки данных для построения экспертной системы. Предполагается, что *правила* находятся в структуре обучающих данных.

Для обучения нейронной сети требуются *обучающие данные*. Они должны отвечать свойствам *представительности и случайности или последовательности*. Все зависит от класса решаемой задачи. Такие данные представляют собой ряды примеров с указанием для каждого из них значением выходного параметра, которое было бы желательно получить. Действия, которые при этом происходят, можно назвать *контролируемым обучением*: «учитель» подает на вход сети вектор исходных данных, а на выходной узел сообщает желаемое значение результата вычислений. Контролируемое обучение нейросети можно рассматривать как решение оптимизационной задачи. Ее целью является минимизация функции ошибок E на данном множестве примеров путем выбора значений весов W . Достижение минимума называется *сходимостью* процесса обучения. Именно возможность этого и доказал Розенблатт. Поскольку ошибка зависит от весов нелинейно, получить решение в аналитической форме невозможно, и поиск глобального минимума осуществляется посредством итерационного процесса - так называемого *обучающего алгоритма*. Разработано уже более сотни разных обучающих алгоритмов, отличающихся друг от друга стратегией оптимизации и критерием ошибок. Обычно в качестве меры погрешности берется средняя квадратичная ошибка (СКО):

$$E = \sqrt{\frac{\sum_{i=1}^M (d_i - y_i)^2}{M}}, \text{ где } M - \text{число примеров в обучающем множестве.}$$

Минимизация величины E осуществляется с помощью градиентных методов. Изменение весов происходит в направлении, обратном к направлению наибольшей крутизны для функции:

$W(t+1) = W(t) + \varepsilon * \frac{\partial E}{\partial W}$, где ε - определяемый пользователем параметр, который называется коэффициентом обучения.

Одним из самых распространенных алгоритмов обучения нейросетей прямого распространения является алгоритм обратного распространения ошибки (BackPropagation, BP). Этот алгоритм был переоткрыт и популяризован в 1986 г. Румельхартом и Мак Клеllandом из группы по изучению параллельных распределенных процессов в Массачусетском технологическом институте.

Основная идея состоит в том, чтобы вычислять чувствительность ошибки сети к изменениям весов. Для этого нужно вычислить частные производные от ошибки по весам. Пусть обучающее множество состоит из P образцов, и входы k -го образца обозначены через $\{x_k^i\}$. Вычисление частных производных осуществляется по *правилу цепи*: вес входа i -го нейрона, идущего от j -го нейрона, пересчитывается по формуле:

$$\Delta W_{ij} = -\varepsilon * \sum_{k=1}^P \frac{\partial E_k}{\partial W_{ij}} = -\varepsilon * \sum_{k=1}^P \frac{\partial E_k}{\partial V_k^i} * \frac{\partial V_k^i}{\partial W_{ij}} = -\varepsilon * \sum_{k=1}^P \delta_k^i * x_k^i,$$

ГДЕ ε - ДЛИНА ШАГА В НАПРАВЛЕНИИ, ОБРАТНОМ К ГРАДИЕНТУ.

Если рассмотреть отдельно k -ый образец, то соответствующее изменение весов равно:

$$\Delta W_{ij} = -\varepsilon * \frac{\partial E_k}{\partial W_{ij}} = -\varepsilon * \delta_k^i * x_k^i.$$

Множитель δ_k^i вычисляется через аналогичные множители из последующего слоя, и ошибка, таким образом, передается в обратном направлении.

Для выходных элементов получим:

$$\delta_k^i = -\frac{\partial E_k}{\partial v_k^i} = -\frac{\partial E_k}{\partial x_k^i} * \frac{\partial x_k^i}{\partial v_k^i} = -(d_k^i - x_k^i) f'(v_k^i).$$

Для скрытых элементов множитель δ_k^i определяется так:

$$\delta_k^i = -\frac{\partial E_k}{\partial v_k^i} = -\sum_h \frac{\partial E_k}{\partial v_k^h} \frac{\partial v_k^h}{\partial v_k^i},$$

где индекс h пробегает номера всех нейронов, на которые воздействует i -ый нейрон.

Алгоритм обратного распространения ошибки представлен на рис. 5.

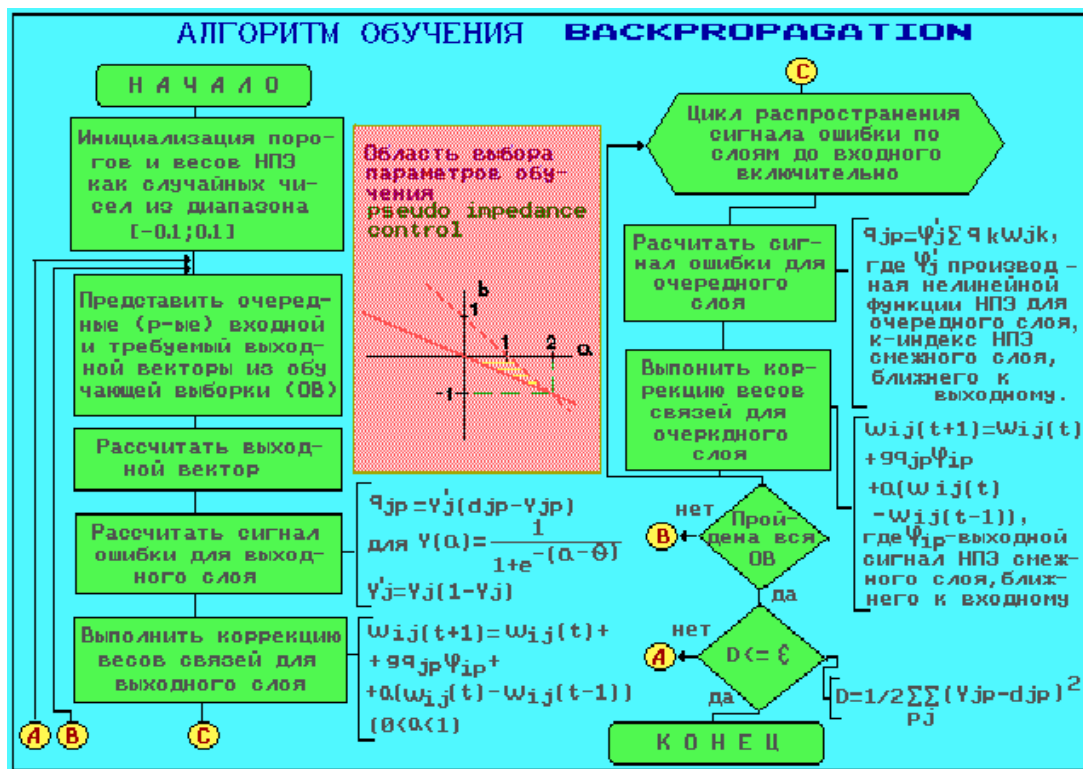


Рис.5 – Алгоритм обучения многослойной нейронной сети

Перед тем, как начинать процесс обучения нейронной сети, необходимо присвоить весам начальные значения. Цель состоит в том, чтобы найти как можно более хорошее начальное приближение к решению и таким образом сэкономить время обучения и улучшить сходимость. Классический подход к этой проблеме состоит в том, чтобы случайным образом выбрать малые значения для всех весов, чтобы быть уверенным, что ни один из сигмоидных элементов не перенасыщен. Однако это не дает полной гарантии, что такое приближение приведет к глобальному минимуму или уменьшит время сходимости. Чтобы обучение не двигалось в ложном направлении при обработке задачи классификации или распознавания, но не задачи аппроксимирования временных рядов, данные нужно перемешивать случайным образом. Иначе нейросеть «выучит» последовательность случайно оказавшихся рядом значений как истинное правило, и потом будет делать ошибку.

Иногда при изменении весов связей нейронов кроме текущего изменения веса к нему прибавляют вектор смещения с предыдущего шага, взятый с некоторым коэффициентом. В этом случае говорят, что учитывается предыдущий импульс движения. Формула изменения веса связи будет выглядеть следующим образом:

$$\Delta W_{ij}(t+1) = \mu * \Delta W_{ij}(t) - (1 - \mu)\varepsilon \frac{\partial E}{\partial W_{ij}}.$$

ГДЕ μ - ЧИСЛО В ИНТЕРВАЛЕ (0,1), КОТОРОЕ ЗАДАЕТСЯ ПОЛЬЗОВАТЕЛЕМ.

Меру точности обучения сети определяет «шаг» изменения параметров в процессе управления обучением. Чем он больше, тем более грубым будет следующее уменьшение суммарной ошибки сети. Чем он меньше, тем больше времени сеть будет тратить на обучение и тем более возможно ее попадание в окрестность локального минимума ошибки. Поэтому управление шагом имеет важное значение для улучшения сходимости нейронной сети. В современных нейросетевых пакетах пользователь может сам определять, как будет изменяться величина шага. Очень часто по умолчанию берется линейная или экспоненциальная зависимость величины шага от количества итераций сети.

Одной из самых больших проблем при использовании нейросетей является невозможность предварительного определения оптимального количества скрытых слоев и нейронов в них. Если нейронов будет слишком мало, то это равносильно потере каких-то нелинейных связей в модели, если нейронов будет много, то это может привести к «переобучению» сети, то есть она просто «выучит» данные, а не распознает их структуру. Поэтому применяется два основных подхода:

- деструктивный подход: берется сеть заведомо большего размера, чем нужно, и в процессе обучения из нее удаляются связи и даже сами нейроны;
- конструктивный подход: первоначально берется маленькая сеть, и к ней, в соответствии со структурой и сложностью задачи, добавляются новые элементы.

Диапазон выходных значений решающих функций нейронов лежит в интервале (0,1) либо (-1,1). Поэтому для лучшей работы сети следует предварительно масштабировать данные обучающей выборки к интервалу от 0 до 1. Это позволит уменьшить ошибки и ускорить функционирование нейросети на этапах обучения и эксплуатации.

Из теоремы об отображении практически любой функции с помощью многослойной нейросети следует, что обучаемая нами нейронная

сет **в принципе** способна сама подстроиться под любые данные с целью минимизации суммарной квадратичной ошибки. Чтобы этого не происходило при обучении нейросетей используют следующий способ проверки сети. Для этого обучающую выборку еще перед началом обучения разбивают случайным образом на две подвыборки: обучающую и тестовую. Обучающую выборку используют собственно для процесса обучения, при этом изменяются веса нейронов. А тестовую используют в процессе обучения для проверки на ней суммарной квадратичной ошибки, но при этом не происходит изменение весов. Если нейросеть показывает улучшение аппроксимации и на обучающей, и на тестовой выборках, то обучение сети происходит в правильном направлении. Иначе может снижаться ошибка на обучающей выборке, но происходить ее увеличение на тестовой. Последнее означает, что сеть «переобучилась» и уже не может быть использована для прогнозирования или классификации. В этом случае немного изменяются веса нейронов, чтобы вывести сеть из окрестности локального минимума ошибки.

Самоорганизующиеся карты. Самоорганизующиеся карты (Self Organizing Maps - SOM) это одна из разновидностей нейросетевых алгоритмов. Основным отличием данной технологии от рассмотренных нами ранее нейросетей, обучаемых по алгоритму обратного распространения, является то, что при обучении используется метод обучения без учителя, то есть результат обучения зависит только от структуры входных данных. Нейронные сети денного типа часто применяются для решения самых различных задач, от восстановления пропусков в данных до анализа данных и поиска закономерностей, например, в финансовой задаче.

Основы самоорганизующихся карт.

Алгоритм функционирования самообучающихся карт представляет собой один из вариантов кластеризации многомерных векторов. Примером таких алгоритмов может служить алгоритм ближайших средних (c-means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов) упорядочены в некоторую структуру (обычно двумерную сетку). При этом в ходе обучения модифицируется не только нейрон-победитель, но и его соседи, но в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. При использовании этого алгоритма вектора, схожие в исходном пространстве, оказываются рядом и на полученной карте. SOM подразумевает использование упорядоченной структуры нейронов. Обычно используются

одно- и двумерные сетки. При этом каждый нейрон представляет собой n -мерный вектор-столбец w_n , где n определяется размерностью исходного пространства (размерностью входных векторов). Применение одно- и двумерных сеток связано с тем, что возникают проблемы при отображении пространственных структур большей размерности (при этом опять возникают проблемы с понижением размерности до двумерной, представимой на мониторе).

Обычно нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте.

При реализации алгоритма SOM заранее задается конфигурация сетки (прямоугольная или шестиугольная), а также количество нейронов в сети. Некоторые источники рекомендуют использовать максимально возможное количество нейронов в карте. При этом начальный радиус обучения (*neighborhood* в англоязычной литературе) в значительной степени влияет на способность обобщения при помощи, полученной карты. В случае, когда количество узлов карты превышает количество примеров в обучающей выборке, то успех использования алгоритма в большей степени зависит от подходящего выбора начального радиуса обучения. Однако в случае, когда размер карты составляет десятки тысяч нейронов, время, требуемое на обучение карты, обычно бывает слишком велико для решения практических задач, таким образом, необходимо достигать допустимого компромисса при выборе количества узлов.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов. Существуют три способа инициирования начальных весов:

1. инициализация случайными значениями, когда всем весам даются малые случайные величины;
2. инициализация примерами, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки;
3. линейная инициализация. В этом случае веса иницируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных. Собственные векторы могут быть найдены, например, при помощи процедуры Грама-Шмидта.

Обучение карт состоит из последовательности коррекций векторов, представляющих собой нейроны. На каждом шаге обучения из исходного набора данных случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве. После того, как найден нейрон-победитель, производится корректировка весов нейрон-сети. При этом вектор, описывающий нейрон-победитель, и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Использование искусственных нейронных сетей в медицине используется в основном в двух направлениях: синтеза классификационных решающих правил для решения диагностических задач (включая прогнозирование возможных состояний больного при различных терапевтических процедурах и технологиях лечения) и прогнозирования значений регистрируемых показателей, характеризующих функционирование как определенной физиологической или функциональной системы так и организма в целом. Разумеется задача прогнозирования качественно разрешается в экстраполяционном временном интервале.

Для решения задачи классификации используются различные пакеты программного обеспечения: Statistica, Matlab Simulink. Neurosolution, специализированные средства для решения определенных задач.

Рассмотрим примеры синтеза нейронных сетей в универсальных оболочках.

1. В пакете STATISTICA:

Для решения задач классификации используется *Мастер решения задач*.

Первый шаг: открывается подготовленный файл данных; ряд переменных (столбцов) являются значениями параметров, категориальная переменная (последний столбец) обозначает диагностический класс.

Второй шаг: открывается окно «*Тип задачи*» («*Problem Type*»); указывается тип задачи и нажимается кнопка «*Next*».

Третий шаг: в окне выбирается зависимая переменная (диагностического класса) и нажимается кнопка «*Next*».

Четвертый шаг: в окне выбираются входные (независимые) переменные; после нажатия кнопки «*Next*» *Мастер решения* автоматически разобьет выборку на обучающую, контрольную и тестовую подвыборки, выделив их черным, синим и красным цветом, соответственно, и производит

перемешивание наблюдений для обеспечения репрезентативности подвыборок.

Пятый шаг: на экране появляется окно «*Длительность поиска*» («*Duration of Design Process*»), в котором задается длительность поиска: быстрый, средний, полный, ограниченный по времени.

Шестой шаг: в открывшемся окне «*Сохранение сетей*» («*Saving Networks*») определяется способ сохранения сетей, например: сохранить сети с лучшим качеством решения, максимальное число сохраняемых сетей и т.д.; затем, в следующем открывшемся окне указываются опции представления результатов.

Седьмой шаг: после нажатия кнопки «*Finish*» STATISTICA производит вычисления и предоставляет итоговый результат в виде таблицы - например, если был выбран на предыдущем шаге режим «сохранить 10 сетей с лучшим качеством», то на экране появятся следующие столбцы: – *Type* – указывается тип сетей, *RBF* – указываются радиальные базисные функции, *Linear* – линейные, *MLP* – многослойный персептрон, ошибка, входы, скрытые, *Perfomance* – качество (указаны доли правильно классифицированных диагностических состояний) – определяется по контрольному подмножеству (доля правильно классифицированных наблюдений – диагностическая эффективность).

Лучшая сеть отмечается знаком «*». Для удаления из набора сетей неудовлетворительного качества используется правая кнопка мыши и команда «*Удалить*» («*Delete*»).

Выделенная нейронная сеть делается активной с помощью команды всплывающего меню «*Выбрать*» («*Select*»).

Если набор нейронных сетей заполнен, то программа *ST Neural Networks* должна определить, какие из имеющихся сетей заменяются вновь создаваемые. После нажатия кнопки *Options-Опции* в диалоговом окне *Редактор набора сетей – Network Set Editor*. На экране появиться окно *Параметры набора сетей – Network Set Options*, в котором задается максимальное количество сетей в наборе (по умолчанию – 30). Установив необходимые значения параметров набора сетей нажимается кнопка *Закрыть – Close*.

Для исследования степени информативности входных переменных в обученной сети применяют *анализ чувствительности* – для этого используют команду *Чувствительность – Sensitiivity* из выпадающего меню. Программа строит таблицу, в которой указывается чувствительность сети по отношению к каждой переменной: ранг, ошибка и отношение. Показатели чувствительности определяются отдельно для обучающего и кон-

трольного набора (подвыборки) наблюдений. В качестве столбцов в окне выступают переменные исходного файла данных.

2. Классификация с помощью персептрона в пакете MatLab.

Классификация состоит из этапов: ввод данных, обучение сети, классификация. Для оценки качества классификации (диагностики) первоначально результаты обследования (и-или результаты мониторинга за состоянием) биообъекта разделяются на две подвыборки: обучающая и тестовая. Если руководствоваться принципом «золотого сечения», то соотношение размеров указанных выборок -0,62:0,38.

Рассмотрим пример 1, реализованный в пакете MatLab 7 SP 2 + Simulink 6, для обучения персептрона:

```
>> % одному из двух классов
>> P=[-0.5 -0.5 +0.3 - 0.1;-0.5 +0.5 -0.5 +1.0];
>> T=[1 1 0 0];
>> plot (P,T);
>> % графическое представление исходных данных
>> % создание персептрона
>> % с указанием границ изменений и одним нейроном
>> My_net=new([1 1; -1 1], 1); % инициализация персептрона
>> % организация цикла адаптивной настройки персептрона
>> % с показом графика линии разделения классов
>> % while (sse(E))
    [My_net, Y,E]=adapt(My_net, P, T);
    Linehandle=plotpc(My_net.IW{1}, My_net.b{1});
    Drawnow;
    End;
```

В приведенной программе исходными данными являются входные векторы с указанием их принадлежности к одному из двух классов – с индексами 0 и 1. P – матрица, столбцы которой соответствуют 4-м входным векторам, T – вектор, элементы которого указывают на принадлежность вектора к нулевому или первому классу. Итоговый результат отображает обучающие вектора и разделяющую их линию, формируемую обученным персептроном.

Для проверки качества созданного классификатора подадим, например, на вход обученного персептрона контрольный вектор $p=\{0.4; 0.5\}$ и проанализируем реакцию сети. Моделирование иллюстрирует следующий программный фрагмент.

```
>> p=[0.4; 0.5];
```



```
>> a = sim(My-net,p);
>>a
a=
0
```

Его работа показывает, что последовала реакция сети $a=0$, т.е., предъявленный контрольный вектор относится к нулевому классу, что является правильным (и графически подтверждается рисунком).

Построение графика реализуется следующим программным фрагментом:

```
>> plot(p,a);
>> hold on;
>> plotpv(P,T);
>> Linehandle=plotpc(My_net.IW{1}, My_net.b{1});
>> hold off
```

Для оценки диагностического качества здесь и далее с помощью статистических вычислений ошибок первого и второго рода классификации необходимо на контрольной выборке для ряда объектов выполнить анализ правильности срабатывания персептрона и рассчитать значения показателей диагностических чувствительности, специфичности и эффективности.

Рассмотрим пример 2, реализованный в пакете MatLab 7 SP 2 + Simulink 6, для обучения **сети со слоем Кохонена**:

Задание: Используя встроенные функции пакета нейронных сетей математической среды Matlab, построить нейронную сеть со слоем Кохонена, которая множество входных данных разделит на кластеры и выявит их центры. На обученную сеть подать новый входной вектор и определить, к какому кластеру он относится.

Для создания нейронной сети со слоем Кохонена воспользуемся встроенной в среду Matlab функцией *newsc*:

```
X=[0 1; 0 1];
clusters=5;
points=5; %Задание количества точек в кластере
std_dev=0.01;
p=nngenc(X,clusters,points,std_dev);%Моделирование входных данных
h=newsc([0 1;0 1],5,.1); % создание слоя Кохонена
h.trainParam.epochs=50; %Задание количества циклов обучения
h=init(h);
h=train(h,p);
```

```

w=h.IW{1}; % вывод графиков исходных данных и выявленных центров
кластеров
plot(p(1,:),p(2:),'^r'),grid;
hold on;
plot(w(:,1),w(:,2),'ob');
xlabel('p(1)');
ylabel('p(2)');
A=0.6
B=0.5
p=[A;B];
plot(A,B,'^k');
y=sim(h,p) % Опрос сети
A =0.6000
B =0.5000
y = (2,1)

```

Результат работы программы отображается в командном окне.

Предъявленный вектор отнесен ко второму кластеру. Теперь данный алгоритм применим к реальной (хоть и элементарной) задаче кластеризации. На вход нейронной сети будем подавать данные весоростовых показателей людей и попробуем выявить три класса (кластера): нормальный весоростовой показатель; избыток веса; недостаток веса. Программный код выглядит следующим образом:

```

%входные данные (первая строка матрицы - рост; вторая - вес)
p=[175 180 182 175 183 176 183 176 183 176 175 180 178 180 178 182 178
182 179 174 172 179; 70 75 100 99 42 48 76 72 40 45 92 96 70 69 95 90 79 82
80 50 96 91] %создаем НС Кохонена с 3 кластерами (нормальный весоростовой
показатель, избыток веса и недостаток веса)
h=newc([0 200;0 100],3,.1);
h.trainParam.epochs=500; %Задание количества циклов обучения
h=train(h,p);
w=h.IW{1};
plot(p(1,:),p(2:),'^r');
hold on;
plot(w(:,1),w(:,2),'ob');
xlabel('Rost');
ylabel('Ves');
% Задание нового входного вектора
% Опрос сети

```

```

A=181
B=65
p=[A;B];
plot(A,B,'+r')
y=sim(h,p)
A =181
B = 65
y = (2,1) 1

```

Результат работы программы представлен в командном окне: $y=(2,1)$
1. Предъявленный вектор отнесен ко второму кластеру.

3. Теперь рассмотрим использование *самоорганизующей карты* на примере двумерных векторов. Используя самоорганизующиеся карты, двумерные векторы разбить на кластеры и выявить их центры, затем подать на вход самоорганизующей карты новый вектор и определить кластер, к которому он относится.

```

P=rand(2,100); %Задание случайных двухмерных входных векторов
figure(1);
hold on;
plot(P(1,:),P(2,:),'+r') %визуальное изображение входных векторов
%Создание НС с 3*4 нейронами
%По умолчанию функция TFCN = 'hextop', то есть нейроны располагаются
в узлах двумерной сетки с шестиугольными ячейками
net=newsom([0 1;0 1],[3 4]);
net.trainParam.epochs=1 %Задание числа циклов настройки
net=train(net,P); % настройка сети
A=0.5
B=0.3
p=[A;B]; % Задание нового входного вектора
plot(A,B,'^k') %прорисовка на рисунке входного вектора (черный тре-
угольник)
figure(2)
plotsom(net.iw{1,1},net.layers{1}.distances)
a=sim(net,p) %опрос сети
A = 0.5000
B = 0.3000
a = (5,1) 1

```

4. Результат работы программы представлен в командном окне: $a = (5,1)$ 1/ Предъявленный вектор отнесен к пятому кластеру.

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал и принципы построения искусственных нейронных сетей, включая язык программирования в MatLab 7 SP 2 + Simulink 6.
2. Сформировать выборки исходных данных по Таблице 1, характеризующие результаты общего анализа крови. Выборки состоят из:
 - для пакета STATISTICA - матрицы, включающей три показателя крови (согласно варианта задания – определяется преподавателем) и идентификаторов состояния («здоров», «болен»), трех матриц типа «показатель», «состояние»;
 - для MatLab 7 SP 2 + Simulink 6 - трех матриц типа «показатель», «состояние».
3. Построить нейронные сети в пакетах STATISTICA и MatLab 7 SP 2 + Simulink 6 (или иных версиях MatLab) – персептрон, сеть Кохонена, самоорганизующие карты для всех сформированных в п.2 матриц.
4. По результатам п.3 сравнить качество работы сетей – в качестве показателя качества использовать диагностическую эффективность.
5. По зафиксированным скрин-шотам работы с программными инструментариями (представляются в отчете по выполнению лабораторной работы) и результатами п.4 сделать выводы о перспективности применения полученных сетей для диагностического процесса.
6. Оформить отчет.

Контрольные вопросы:

1. Что понимается под кластеризацией?
2. Для чего применяются искусственные нейронные сети?
3. Каким образом классифицируются искусственные нейронные сети?
4. Для решения каких задач предназначены искусственные нейронные сети?
5. В чем заключается закон обучения Кохонена?
6. Опишите встроенные операторы Matlab для кластеризации.
7. Зачем используются самоорганизующиеся карты? Чем отличаются сети Кохонена от SOM?

8. Как устроен персептрон?
9. В чем заключается принцип обратного распространения ошибки в искусственных нейронных сетях?
10. Что такое – «скрытые слои» искусственной нейронной сети?
11. Как проверяется диагностическое качество работы искусственной нейронной сети?
12. Перечислите основные недостатки и преимущества применения искусственных нейронных сетей в диагностическом процессе?
13. В чем состоят отличия искусственных нейронных и иммунных сетей?

Таблица 1. Данные для построения искусственной нейронной сети.

Больные					
№ п.п.	эритроци- ты	гемоглобин	Цветовой показатель	моноциты	СОЭ
1	4,6	142	0,93	2	9
2	4,2	115	0,82	2	16
3	3,4	107	0,94	2	8
4	5	170	1,02	4	2
5	3,6	105	0,88	2	22
6	5,3	108	0,61	2	6
7	4,16	132	0,95	4	10
8	4,3	145	1,01	3	4
9	5	160	0,96	1	4
10	3,1	108	1,05	4	12
11	4	124	0,93	3	10
12	5,1	164	0,96	6	12
13	4,25	138	0,97	4	26
14	4,3	150	1,05	9	9
15	4,8	153	0,96	2	9
16	3,6	114	0,95	1	14
Здоровые					
1	4,2	137	0,98	3	3
2	4,05	125	0,93	2	7
3	3,9	125	0,96	9	5
4	3,85	120	0,94	5	5
5	4,2	134	0,96	8	4
6	3,65	117	0,96	3	9
7	4,4	146	0,995	4	3
8	4	122	0,92	2	10
9	4,7	150	0,96	3	6
10	4,6	144	0,94	6	3
11	4,26	140	0,99	6	3
12	4,2	140	1	6	6
13	4,5	140	0,93	9	1
14	4,5	150	1	4	4
15	4,4	148	1,01	10	2
16	4,4	143	0,98	8	4

ЛАБОРАТОРНАЯ РАБОТА №3: ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ ТИПОВЫХ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ.

Цель работы: Исследовать поведение интегральной функции принадлежности с точки зрения темпа ее сходимости к определенной асимптоте в зависимости от методики ее вычисления и ее составляющих.

Краткие теоретические сведения:

Нечеткие множества – это частный случай нечисловых данных.

Пусть A - некоторое множество. Подмножество B множества A характеризуется своей характеристической функцией:

$$\mu_B(x) = \begin{cases} 1, & x \in B, \\ 0, & x \notin B. \end{cases} \quad (1)$$

Что такое нечеткое множество? Обычно говорят, что нечеткое подмножество C множества A характеризуется своей функцией принадлежности $\mu_C: A \rightarrow [0;1]$. Значение функции принадлежности в точке x показывает степень принадлежности этой точки нечеткому множеству. Нечеткое множество описывает неопределенность, соответствующую точке x – она одновременно и входит, и не входит в нечеткое множество C . За вхождение - $\mu_C(x)$ шансов, за второе, т.е. за то, что точка не входит в множество, $(1 - \mu_C(x))$ шансов.

Если функция принадлежности $\mu_C(x)$ имеет вид (1) при некотором B , то C есть обычное (четкое) подмножество A . Таким образом, теория нечетких множеств является более общей или хотя бы не менее общей математической дисциплиной, чем обычная теория множеств, поскольку обычные множества – частный случай нечетких. Соответственно можно ожидать, что теория нечеткости как целое обобщает классическую математику.

По степени общности обычная математика и нечеткая математика эквивалентны. Однако для практического применения, например, в теории принятия решений описание и анализ неопределенностей с помощью теории нечетких множеств весьма плодотворны.

Теория нечеткости является обобщением интервальной математики. Действительно, функция принадлежности:

$$\mu_B(x) = \begin{cases} 1, & x \in [a; b], \\ 0, & x \notin [a; b] \end{cases}$$

задает интервальную неопределенность – про рассматриваемую величину известно лишь, что она лежит в заданном интервале $[a,b]$. Тем самым описание неопределенностей с помощью нечетких множеств является более общим, чем с помощью интервалов.

Л.А. Заде рассматривал теорию нечетких множеств как аппарат анализа и моделирования гуманистических систем, т.е. систем, в которых участвует человек. Его подход опирается на предпосылку о том, что элементами мышления человека являются не числа, а элементы некоторых нечетких множеств или классов объектов, для которых переход от «принадлежности» к «непринадлежности» не скачкообразен, а непрерывен. В настоящее время методы теории нечеткости используются почти во всех прикладных областях, в том числе при управлении предприятиями, качеством продукции и технологическими процессами, при описании предпочтений и оптимизации различных процессов, в том числе диагностического характера.

Л.А. Заде использовал термин «fuzzy set» (нечеткое множество). На русский язык термин «fuzzy» переводится как нечеткий, размытый, расплывчатый, и даже как пушистый и туманный.

В качестве примера аппарата теории нечетких множеств рассмотрим определения теоретико-множественных операций над нечеткими множествами. Пусть C и D - два нечетких подмножества A с функциями принадлежности $\mu_C(x)$ и $\mu_D(x)$ соответственно. Пересечением $C \cap D$, произведением CD , объединением $C \cup D$, отрицанием \bar{C} , суммой $C+D$ называются нечеткие подмножества A с функциями принадлежности, соответственно:

$$\mu_{C \cap D}(x) = \min(\mu_C(x), \mu_D(x)), \quad \mu_{CD}(x) = \mu_C(x)\mu_D(x), \quad \mu_{\bar{C}}(x) = 1 - \mu_C(x),$$

$$\mu_{C \cup D}(x) = \max(\mu_C(x), \mu_D(x)), \quad \mu_{C+D}(x) = \mu_C(x) + \mu_D(x) - \mu_C(x)\mu_D(x), \quad x \in A.$$

Теория нечетких множеств в определенном смысле сводится к теории случайных множеств. Однако при решении прикладных задач вероятностно-статистические методы и методы теории нечеткости

Законы де Моргана для нечетких множеств. Законами же Моргана называются следующие тождества алгебры множеств:

$$\overline{A \cup B} = \bar{A} \cap \bar{B}, \quad \overline{A \cap B} = \bar{A} \cup \bar{B}. \quad (2)$$

$$\overline{A \cup B} = \bar{A} \cap \bar{B}, \quad \overline{A \cap B} = \bar{A} \cup \bar{B}, \quad (3)$$

$$\overline{A+B} = \bar{A}\bar{B}, \quad \overline{\bar{A}\bar{B}} = \bar{A} + \bar{B}. \quad (4)$$

Тождества (3) и (4) называются *законами де Моргана для нечетких множеств*.

Дистрибутивный закон для нечетких множеств. Некоторые свойства операций над множествами не выполнены для нечетких множеств. Так, $A + A \neq A$, за исключением случая, когда A – «четкое» множество (т.е. функция принадлежности принимает только значения 0 и 1).

Носителем нечеткого множества A называется совокупность всех точек $y \in Y$, для которых $\mu_A(y) > 0$.

Нечетким логическим выводом (fuzzy logic inference) называется получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций.

В общем случае нечеткий вывод решения происходит за три (или четыре) шага:

1) *этап фаззификации*. С помощью функций принадлежности всех термов входных лингвистических переменных и на основании задаваемых четких значений из универсумов входных лингвистических переменных определяются степени уверенности в том, что выходная лингвистическая переменная принимает конкретное значение.

2) *этап непосредственного нечеткого вывода*. На основании набора правил - нечеткой базы знаний - вычисляются значения истинности для предпосылок всех правил на основании конкретных нечетких операций, соответствующих конъюнкции или дизъюнкции термов в левой части правил. В большинстве случаев это либо максимум, либо минимум из степеней уверенности термов, вычисленных на этапе фаззификации, который применяется к заключению каждого правила. Используя один из способов построения нечеткой импликации, получается нечеткая переменная, соответствующая вычисленному значению степени уверенности в левой части правила и нечеткому множеству в правой части правила.

Основным способом построения нечеткой импликации является способ Kleene-Dienes: пусть A и B - нечеткие высказывания и μ_A, μ_B — соответствующие им функции принадлежности. Импликация $A \Rightarrow B$ определяется формулами:

$$A \Rightarrow B \equiv \bar{A} \vee B,$$

$$\mu_{A \Rightarrow B}(x, y) = \max(1 - \mu_A(x); \mu_B(y)).$$

Однако, это не единственное обобщение оператора импликации. В таблице приведены другие интерпретации этого понятия, наиболее часто применяемые на практике.

Larsen	$\mu_{A \Rightarrow B}(x, y) = \mu_A(x) \mu_B(y)$
Lukasiewicz	$\mu_{A \Rightarrow B}(x, y) = \min(1; 1 - \mu_A(x) + \mu_B(y))$
Mamdani	$\mu_{A \Rightarrow B}(x, y) = \max(\mu_A(x); \mu_B(y))$
Kleene-Dienes-Lu	$\mu_{A \Rightarrow B}(x, y) = 1 - \mu_A(x) + \mu_A(x) \mu_B(y)$

3) *этап композиции (агрегации, аккумуляции)*. Все нечеткие множества, назначенные для каждого термина каждой выходной лингвистической лингвистической переменной, объединяются вместе, и формируется единственное нечеткое множество - значение для каждой выводимой лингвистической переменной. Обычно используются функции MAX или SUM.

4) *этап дефаззификации (необязательный)*. Используется тогда, когда полезно преобразовать нечеткий набор значений выводимых лингвистических переменных к точным. В теории нечетких множеств процедура дефаззификации аналогична нахождению характеристик положения (математического ожидания, моды, медианы) случайных величин в теории вероятности. Имеется достаточно большое количество методов перехода к точным значениям.

Простейшим способом выполнения процедуры дефаззификации является выбор четкого числа, соответствующего максимуму функции принадлежности. Однако пригодность этого способа распространяется лишь на одноэкстремальные функции принадлежности. Для многоэкстремальных функций принадлежности часто используются следующие методы дефаззификации:

1) COG (Center Of Gravity) – «центр тяжести». Физическим аналогом этой формулы является нахождение центра тяжести плоской фигуры, ограниченной осями координат и графиком функции принадлежности нечеткого множества.

2) MOM (Mean Of Maximums) – «центр максимумов». При использовании метода центра максимумов требуется найти среднее арифметическое элементов универсального множества, имеющих максимальные степени принадлежности.

3) First Maximum – «первый максимум» - максимум функции

принадлежности с наименьшей абсциссой.

Конкретные способы синтеза интегральных функций принадлежности определяется из характера решаемой задачи, ее целей и предметной области.

В связи с этим, на этапе разведочного анализа результатов мониторинга или в процессе диагностики целесообразно проводить исследование зависимости результирующей функции принадлежности от значений функции принадлежностей аргументов в нее входящих.

Порядок выполнения работы:

1. Изучить теоретические сведения - теорию нечетких множеств, принципы выбора и построения функций принадлежности, методы вычисления интегральной функции принадлежности.
2. На абстрактном носителе x синтезировать функции принадлежности согласно Таблице 1.
3. Согласно Таблице 2 составить алгоритм вычисления интегральной функции принадлежности в случае применения N решающих правил обладающих определенными функциями принадлежностями.
4. Оценить значения скорости и ускорения сходимости интегральной функции к единице в зависимости от количества решающих правил и выбранной функции принадлежности.
5. Проанализировать полученные результаты с точки зрения: сходимости, применения определенных функций принадлежности и методов вычисления интегральной функции, количества итераций в процессе вычисления.
6. Сделать выводы.
7. Оформить отчет, включающий в себя результирующую таблицу вида:

	Способ получения интегрального показателя
Функция принадлежности	Скорость, ускорение, график

8. В процессе самостоятельной работы в ходе выполнения лабораторной работы рекомендуется проанализировать показатели вычисления интегральной функции принадлежности в случаях изменения в итерациях как правил вычисления функций принадлежности, так и правил вычисления интегральной функции.

Таблица 1. – Типовые функции принадлежности

Название функции	Вид функции
S	$s(x;a,b,c) = \begin{cases} 0, \text{ для } x \leq a \\ 2 \cdot \left(\frac{x-a}{c-a}\right)^2, \text{ для } a \leq x \leq b \\ 1 - 2 \cdot \left(\frac{x-c}{c-a}\right)^2, \text{ для } b \leq x \leq c \\ 1, \text{ для } x \geq c \end{cases}$
π	$\pi(x;b,c) = \begin{cases} s(x;c-b, c-b/2, c), \text{ для } x \leq c \\ 1 - s(x;c, c+b/2, c+b), \text{ для } x \geq c \end{cases}$
γ	$\gamma(x;a,b) = \begin{cases} 0, \text{ для } x \leq a \\ \frac{x-a}{b-a}, \text{ для } a \leq x \leq b \\ 1, \text{ для } x \geq b \end{cases}$
t	$t(x;a,b,c) = \begin{cases} 0, \text{ для } x \leq a \\ \frac{x-a}{b-a}, \text{ для } a \leq x \leq b \\ \frac{c-x}{c-b}, \text{ для } b \leq x \leq c \\ 0, \text{ для } x \geq c \end{cases}$
L	$L(x;a,b) = \begin{cases} 1, \text{ для } x \leq a \\ \frac{b-x}{b-a}, \text{ для } a \leq x \leq b \\ 0, \text{ для } x \geq b \end{cases}$

Таблица 2 Интегральные функции принадлежности (пример)

№	T(a,b) – «пересечение»	S(a,b) - «объединение»	параметры
1	Min(a,b)	Max(a,b)	
2	a*b	a+b-a*b	
3	MAX(a+b-1,0)	Min(a+b,1)	
4	$\begin{cases} a, & b=1 \\ b, & a=1 \\ 0, & \text{èí à ÷ à} \end{cases}$	$\begin{cases} a, & b=0 \\ b, & a=0 \\ 0, & a, b \neq 0 \end{cases}$	
5	$\frac{a \cdot b}{\gamma + (1-\gamma) \cdot (a+b-a \cdot b)}$	$\frac{a+b-(2-\gamma) \cdot a \cdot b}{1-(1-\gamma) \cdot (a \cdot b)}$	$\gamma > 0$

6	$\frac{a \cdot b}{\max(a, b, \alpha)}$	$\frac{a + b - a \cdot b - \min(a, b, 1 - \alpha)}{\max(1 - a, 1 - b, \alpha)}$	$\alpha \in [0, 1]$
7	$1 - \sqrt[p]{(1 - a)^p + (1 - b)^p - (1 - a)^p \cdot (1 - b)^p}$	$\sqrt[p]{a^p + b^p - a^p \cdot b^p}$	$p > 0$
8	$[1 + \sqrt[\lambda]{(\frac{1}{a} - 1)^\lambda + (\frac{1}{b} - 1)^\lambda}]^{-1}$	$[1 + \sqrt[\lambda]{(\frac{1}{a} - 1)^{-\lambda} + (\frac{1}{b} - 1)^{-\lambda}]^{-1}$	$\lambda > 0$
9	$\max(1 - \sqrt[p]{(1 - a)^p + (1 - b)^p}, 0)$	$\min(\sqrt[p]{a^p + b^p}, 1)$	$p \geq 1$
10	$\log_w(1 + \frac{(w^a - 1) \cdot (w^b - 1)}{w - 1})$	$1 - \log_w(1 + \frac{(w^{1-a} - 1) \cdot (w^{1-b} - 1)}{w - 1})$	$w = \min(1, 1 - a; 1, 1 - b)$
12	$\max(\frac{a + b - 1 + \lambda \cdot a \cdot b}{1 + \lambda}, 0)$	$\min(a + b + \lambda \cdot a \cdot b, 1)$	$\lambda \geq -1$

Контрольные вопросы:

1. Чем отличается нечеткое множество от четкого?
2. Что такое функция принадлежности?
3. Как получить интегральное значение функции принадлежности в случае анализа нескольких нечетких множеств?
4. Что такое носитель функции принадлежности?
5. Когда рекомендуется применять аппарат теории нечетких множеств при построении диагностических правил?
6. В чем заключаются алгоритмы нечеткого вывода Мамдани, Лоренса, Цукамото?
7. Что такое лингвистическая переменная?
8. Каким образом осуществляется коррекция интегральной функции принадлежности в случае неадекватно быстрого приближения ее значения к 1?

ЛАБОРАТОРНАЯ РАБОТА №4: ЭКСПЕРТНАЯ СИСТЕМА ДИФФЕРЕНЦИАЛЬНОЙ ДИАГНОСТИКИ ЗАБОЛЕВАНИЙ.

Цель работы: изучение принципов работы, проектирования и овладение навыками эксплуатации экспертной системы диагностики заболеваний.

Краткие теоретические сведения.

Успехи врачевания в первую очередь определяются качеством распознавания болезней. В связи с этим правомочен афоризм: «Правильная диагностика – хорошее лечение».

Непрерывный научно-технический прогресс во всех научных областях, в том числе и в медицине, дает для расширения диапазона диагностических возможностей врача множество современных методов обследования больного. Современные методы исследования болезни позволяют более глубоко и детально изучить их, способствуют открытию новых патологических состояний, дают возможность уточнить этиопатогенетические механизмы их возникновения. Однако качество диагноза определяется не только открытием и внедрением инструментально-технических и других методов распознавания болезней в клиническую медицину, но и умением практических врачей мыслить. С увеличением информации о болезнях усложняется процесс мыслительной деятельности врача, повышаются требования к логической обоснованности, своевременности, завершенности и точности установления диагноза. Проблема удовлетворения современным требованиям, предъявляемым к диагнозу, является самой главной, определяющей задачей врача, которая реализуется с помощью методической части клинической медицины – диагностики - науки о распознавании болезней.

• ***Диагностика*** – раздел медицинской науки изучающей методы распознавания болезней в процессе исследования, наблюдения больного и мышления врача с целью установления диагноза

Решая задачу распознавания болезни, врач имеет несколько возможностей, использование которых обеспечивает высокое качество диагностики:

- основательный уровень владения методами и техникой врачебного исследования;
- внедрение в клиническую медицину новых современных методов исследования;
- совершенствование и углубление знаний о признаках болезни и их диагностической ценности;

- совершенствование мыслительных операций с выявленными признаками болезни научно обоснованными способами диагностики;
- совершенствование и разработка методов диагностических решений (создание диагностического алгоритма, схем эвристического поиска, диагностических тестов и др.).

Анализ клинической картины начинается с процесса обследования больного. Важно понимать, что обследование – отнюдь не механическое, а творческое занятие. Его творческий компонент неразрывно связан с глубиной характеристики симптома. Группировка и классификация симптомов являются основными моментами анализа клинической картины. Процедура систематизации подразумевает группировку симптомов по системам органов. В этом процессе проверяется полнота обследования больного и вырабатывается привычка к строгому, последовательному обследованию и изложению обнаруженных фактов.

Компоненты синдрома – симптомы, полученные при сборе анамнеза и физикальных исследований. Синдром синтезируется из симптомов, с целью локализовать зону поражения и выявить патологию. Например, установленное при обследовании системы дыхания локализованное притупление перкуторного звука, бронхиальное дыхание и усиление бронхофонии свидетельствуют об уплотнении легочной ткани. Важнейшая предпосылка успеха диагноза – глубокий патогенетический анализ клинической ситуации.

Дифференциальная диагностика – это сравнение клинической картины заболевания у данного конкретного больного со всеми заболеваниями, имеющими сходную клиническую картину, для исключения всех болезней, кроме одной.

Это сравнение должно проводиться не хаотично, не стихийно, а по строгим правилам и в определенном порядке. Иными словами, мыслительный процесс при дифференциальной диагностике должен подчиняться такой же строгой схеме, как, например, проведение физикального исследования или анализ электрокардиограммы. Практически все авторы, занимавшиеся этой проблемой сходятся, с небольшими отличиями, в том мнении, что проведение дифференциальной диагностики состоит из следующих пяти этапов:

- группировка симптомов для выделения синдромов;
- выделение ведущего синдрома;
- генерация диагностической гипотезы;
- построение дифференциально-диагностического ряда;
- исключение синдромно-сходных заболеваний и формулировку окончательного диагноза.

чательного диагноза.

Рассмотрим последовательность действий на каждом этапе проведения дифференциальной диагностики.

Первый этап – группировка симптомов для выделения синдромов, был достаточно описан выше, но хотелось бы подчеркнуть, что при выделении синдрома следует оценить его особенности. Подробная патогенетическая характеристика синдромов абсолютно необходима для плодотворной деятельности на следующих этапах.

На втором этапе ведущий синдром выделяется по тем же критериям, что и ведущий симптом. Главное здесь – плодотворность генерации диагностической гипотезы.

На третьем этапе формируется диагностическая гипотеза. Диагностическая гипотеза – это, в сущности, рабочий диагноз, который создается методом аналогии на основе синдромного анализа.

Характерными чертами диагностической гипотезы являются:

- неполнота аргументации;
- уникальность диагностической гипотезы.

Неполнота аргументации означает что рабочий диагноз формулируется у постели больного на базе данных анамнеза и физикального обследования. Следует иметь в виду, что как анамнестическое, так и физикальное обследование, несмотря на стереотипизм их реализации, не предусматривают некоторых вопросов, приемов, методов, которые имеют существенное значение для диагностики относительно редких заболеваний. Например, проведение провокационных проб для диагностики кровотоковости петехиально-пятнистого типа, оценка ширины пространства между относительной и абсолютной тупостью сердца для верификации гидроперикарда и т.д.

Под уникальностью диагностической гипотезы в данном случае, имеется в виду то, что диагностическая гипотеза генерируется в процессе работы с каждым больным. Она может звучать одинаково, при работе с довольно большим количеством больных, например «гипертоническая болезнь», но аргументация в пользу этого предположения каждый раз создается заново.

Считается, что в процессе генерации диагностической гипотезы возможен ряд ошибок логического порядка:

- ошибка поспешного обобщения;
- ошибка в умозаключении по аналогии;
- ошибка ложной последовательности.

Ошибкой поспешного обобщения является учет диагностической гипотезой не всех, а лишь некоторых клинических данных. Ошибка в умозаключении по аналогии происходит, когда обобщаются процессы, имеющие разный патогенез. Ошибка ложной последовательности возможна, если умозаключение движется по пути «после этого, значит вследствие этого», тогда как далеко не всегда последовательное возникновение симптомов означает наличие между ними причинно – следственной связи.

На четвертом этапе строится дифференциально - диагностический ряд, что требует от врача широкой эрудиции с одной стороны и навыков системного мышления с другой. Следует обратить особое внимание, что дифференциально - диагностический ряд - не простое перечисление заболеваний, имеющих один отличительный признак (выделенный в качестве ведущего синдрома), на большие или меньшие группы. Разделение это осуществляется на патогенетической основе.

Такого рода внутрисиндромные классификации чрезвычайно полезны, т.к. позволяют в дальнейшем проводить дифференциальную диагностику, исключая заболевания не по одному, а целыми группами. Работа по составлению дифференциально-диагностического ряда сопровождается его написанием. Это служит гарантией того, что ни одно из заболеваний не будет забыто в процессе дифференциальной диагностики.

На пятом этапе осуществляется процесс исключения синдромно - сходных заболеваний в два этапа. На первом фазе производятся межгрупповые сопоставления и исключение всех групп заболеваний, кроме одной. На второй фазе происходит сопоставление заболеваний внутри оставшейся группы и исключение всех болезней, кроме одной.

Определяются пять основных принципов дифференциальной диагностики.

Первый принцип – принцип существенного различия.

Принцип существенного различия означает, что наблюдаемый случай болезни не принадлежит к сравниваемому с ним виду заболеваний, т.к. в клинической картине у данного больного отсутствует синдром, обязательно имеющий место при сравниваемой болезни, или группе болезней. Принцип существенного различия имеет и другую трактовку: заболевание не относится к сравниваемому с ним виду, т.к. у больного имеется синдром (симптом), никогда не встречающийся при сравниваемом заболевании.

Второй принцип – исключения через противоположность. Этот принцип можно сформулировать так: у больного имеется симптом,

прямо противоположный тому, который закономерно выявляется при сравниваемой болезни.

Третий принцип – принцип несовпадения признаков – важнейший и наиболее часто используемый. В данном случае речь идет о существенном сходстве симптома и в то же время – о его отличительных характеристиках при различных заболеваниях.

Четвертый принцип – индуктивного дифференциального диагноза -предусматривает постоянную работу в библиотеке. По прочтении соответствующей предполагаемому заболеванию литературы неизбежно возникает потребность повторить обследование больного с фиксацией внимания на специфических проявлениях болезни.

Пятый принцип – проверки диагноза. В зависимости от случая, проверка диагноза может быть либо очень короткой (в самых простых случаях она не нужна), либо продолжительной, трудной для врача и больного. Иногда приходится двигаться от простых методов к все более и более сложным, если простые и необременительные способы диагностики на момент их использования не дают определенных результатов.

Применение методики дифференциальной диагностики по одному ведущему синдрому плодотворна, но иногда, особенно при сложных, полисиндромных клинических картинах моно-синдромный дифференциальный диагноз не приводит к успеху. Диагностическая гипотеза (рабочий диагноз) не оказывается ни подтвержденной, ни отвергнутой. В этой ситуации все приходится начинать сначала, выделяя другой синдром в качестве основного. При безуспешности этой попытки следует перейти к дифференциальной диагностике по третьему синдрому и т.д.

Вообще при полисиндромных процессах иногда целесообразно взглянуть на вещи более широко: все многообразие клиники может быть обусловлено не одним заболеванием, а их сочетанием.

После сбора анамнеза, проведения физикального исследования с целью проверки диагностических версий, врач использует дополнительные специальные диагностические тесты, применяя лабораторные и инструментальные методы исследования.. Диагностический тест необходим для подтверждения или исключения болезни .

В зависимости от цели выделяют специальные методы исследования – инвазивные и неинвазивные.. Выделяют также следующие методы исследования :

- функциональные методы, оценивающие ту или иную функцию органа или системы;
- индикаторные, указывающие на наличие какого-либо патоло-

гического процесса или состояния (воспаление, цитолиз, иммунные нарушения и т. д.);

– визуализирующие методы, позволяющие оценить макроструктуру органа;

– морфологические — прижизненно оценивающие микроструктуру органа.

При дополнительном обследовании больного необходимо учитывать чувствительность, специфичность и информативность используемых методов. Информативность исследования — неколичественная характеристика метода, отражающая объем полезной информации, получаемой при применении данного метода в конкретной патологии и, следовательно, позволяющей судить, насколько данный метод необходим в конкретном случае.

Наибольшую ценность в диагностике имеют диагностические тесты высокой чувствительности и специфичности, позволяющие обнаруживать высокоспецифические и патогномоничные диагностические стигматы.

Оценка надежности различных диагностических методов (тестов) проводится в специальных научных исследованиях, где изучаемый метод сравнивают с эталонным. Однако, прежде чем определить критерии надежности результатов диагностического теста, необходимо оценить априорную вероятность наличия болезни.

Она особенно важна в тех случаях, когда чувствительность и специфичность применяемого метода низкая, а эталонный тест либо отсутствует, либо его трудно осуществить. Оценка априорной вероятности основывается на данных эпидемиологических характеристик болезни: степени ее распространенности, особенности ее течения в той или иной местности, факторах риска и степени соответствия клинической картины данного больного классическому описанию болезни. В практической деятельности врач, оценивая априорную вероятность болезни ограничивается приблизительным количественным ориентиром. Эта ориентировочная, приблизительная, количественная оценка априорной вероятности выводится на основании знаний литературных источников, подсказанной интуиции, клинических наблюдений и накопленного опыта.

Вероятность болезни, определяемая по результатам проведения диагностического теста, называется апостериорной вероятностью.

Результат теста может быть либо положительным (патология), либо отрицательным (норма), а заболевание может либо присутствовать, либо отсутствовать. Возможны четыре варианта толкования результатов теста — два истинных и два ложных. Правильный ответ — это положительный

результат, если заболевание есть, или отрицательный, если заболевание отсутствует. Однако результаты могут быть и ложноотрицательными (болезнь есть, но тест ее исключает), и ложноположительными (болезни нет, но тест ее подтверждает) (таблицы 1,2). Оценка точности теста основана на сравнении его результата с критерием, заслуживающим доверия истины, - эталоном. С этой целью определяют чувствительность и специфичность теста, предсказательную ценность положительного и отрицательного теста, а также диагностическую точность метода.

Таблица 1 - Варианты оценки результатов теста

Тесты	Болезнь есть	Болезни нет
Положительный	A - истинноположительный	B - ложноположительный
Отрицательный	C - ложноотрицательный	D - истинноотрицательный

Таблица 2 - Оценка точности результатов теста

Чувствительность теста	$A / (a + c)$ - доля истинно положительных тестов среди больных
Специфичность теста	$D / (b + d)$ - доля истинно отрицательных тестов среди здоровых
Предсказанная ценность положительного теста	$A / (a + b)$ = доля истинно положительных тестов среди всех положительных тестов
Предсказанная ценность отрицательного теста	$D / (c + d)$ = доля истинно отрицательных тестов среди всех отрицательных тестов
Диагностическая точность метода	$(a + d) / (a + b + c + d)$ = доля истинных результатов среди всех результатов теста

Поскольку точные способы установления диагноза почти всегда трудоемкие, дорогие и сопряжены с риском исследования, оправдано назначение более простых и доступных тестов, особенно на ранних эта-

пах исследования больного. Вероятность положительного результата диагностического теста в присутствии болезни называется чувствительностью метода, а вероятность отрицательного результата в отсутствие болезни – его специфичностью. При выборе диагностического теста врач должен принимать во внимание как чувствительность, так и его специфичность.

Чувствительность метода для выявления конкретного признака определяется соотношением количества больных с наличием признака по данным применяемого метода исследования и количества обследованных больных с действительным наличием этого признака, выраженным в процентном отношении. Если метод или тест обладает высокой чувствительностью, то он выявляет патологию у большинства больных с данным заболеванием (истинно положительные результаты) и лишь у незначительного количества больных дает ложноотрицательные результаты. Чувствительность признака при конкретном заболевании – показатель, отражающий частоту встречаемости признака при данной патологии. Специфичность метода для выявления конкретного признака – соотношение количества обследованных здоровых пациентов, у которых признак отсутствует, и общего количества здоровых пациентов, выраженное в процентном отношении. Таким образом, специфичность отражает частоту «ошибок» метода исследования у здоровых людей. Специфичность признака при данном заболевании – показатель, отражающий насколько частый признак бывает у людей, не имеющих этого заболевания. Чувствительный тест следует убрать, если есть риск пропустить опасную, но излечимую болезнь. Чувствительные тесты также рациональны на ранних стадиях диагностического поиска с целью сужения круга вероятных заболеваний].

Диагностический метод безупречен, если и чувствительность и специфичность равны 100 %. Однако таких совершенных методов исследования мало или они труднодоступны. Если проводится эталонный тест, при котором чувствительность и специфичность равны 100 %, определение априорной вероятности подозреваемой болезни почти не играет роли. И наоборот, чем меньше чувствительность и специфичность, тем значительнее предварительная информация о больном.

Таким образом, в одних случаях решающим фактором для интерпретации результатов диагностического теста будет как можно более точная оценка априорной вероятности, т. е. вероятности еще до проведения диагностического теста, а в других определяющее значение в верификации болезни будет иметь эталонный тест или тест с высокой чувствительностью и его специфичностью.

При положительном результате диагностического теста апостериорная вероятность болезни идентифицирует предсказательную ценность

Из данных таблицы 3 видно, что апостериорная вероятность болезни значительно зависит от оценки априорной вероятности. Следовательно, при оценке эффективности положительных результатов для подтверждения диагноза и отрицательных – для его исключения необходимо учитывать вероятность подозреваемой болезни у обследуемого больного еще до проведения диагностического теста, особенно, если диагностические тесты несовершенны (низкой чувствительности и специфичности).

Таблица 3 - Пример диагностического теста

Априорная вероятность	2%		30%		70%	
	Есть	Нет	Есть	Нет	Есть	Нет
Болезнь	Есть	Нет	Есть	Нет	Есть	Нет
Чувствительность и специфичность	ДЧ	ДС	ДЧ	ДС	ДЧ	ДС
Положительный тест	18	196	270	140	630	90
Отрицательный тест	2	784	30	560	70	210
Всего	20	980	300	700	700	300
Предсказанная ценность положительного теста	8,4% 18/(18+196)		65,85% 270/(270+140)		87,5% 630/(630+90)	

В таблице ДЧ и ДС – диагностические чувствительность и специфичность, соответственно.

Считается, что вероятность болезни тем выше, чем больше свидетельств в ее пользу. Поэтому, если результат диагностического теста сомнительный, необходимо полагаться на сумму доказательств. Но эти данные повышают диагностические возможности только в том случае, если назначаемые диагностические тесты дают независимые друг от друга информативные доказательства.

Экспертные системы дифференциальной диагностики предназначены для решения следующих основных задач: интерпретация, прогноз, диагностика, проектирование, планирование, обучение, наблюдение, управление.

Экспертные системы имеют в своем составе обширную базу данных – факты выбранной предметной области, а также базу знаний, в которой отражены профессиональные навыки и умения специалистов высокого уровня в данной области.

В качестве примера российской разработки можно привести комплекс РОФЭС (Регистратор Оценки Функционально-Эмоционального Состояния). Система состоит из комплекса датчиков и программного обеспечения, с помощью которых производится диагностика и анализ состояния организма. Она может осуществить подбор препаратов для конкретного человека, провести частотную и цветотерапию (с использованием обратной связи), определить гормональный профиль и риск опасных видов заболеваний. Таким образом, РОФЭС производит не только оценку, но и управление состоянием здоровья.

В качестве примера современных экспертных систем, использующихся в медицинских учреждениях, можно привести следующие:

1 Экспертная система ДИН. С целью повышения качества диагностики и эффективности лечения критических состояний у детей в Московском НИИ педиатрии и детской хирургии создана экспертная система ДИН, ориентированная на решение задачи распознавания неотложного состояния у детей в терминах синдрома или нескольких синдромов (точнее, характеризующих их состояний, которые отражают степень выраженности синдрома) при предъявлении признаков заболеваний, под которыми следует понимать анамнестические, клинические и лабораторные проявления. База знаний ЭС содержит описания 34 синдромов, которые включают 84 состояния. Для системы это список диагностических предположений-гипотез. База экспертных знаний врача-реаниматолога содержит более 1000 диагностических критериев и заключений о динамике развития неотложного состояния. С целью ускорения работы системы, ориентированной на угрожающие жизни состояния, все множество синдромов подразделено на 14 групп в соответствии с преобладающими нарушениями со стороны той или иной системы жизнеобеспечения, и каждая группа имеет свои дифференцирующие симптомы. Знания о синдромах представлены таким образом, что охватывают: условие выдвижения гипотезы о возможности возникновения синдрома, которое является некоторой логической комбинацией симптомов; клиническую картину, то есть симптомы заболевания; необходимые и достаточные условия для подтверждения гипотезы; информацию о дополнительных синдромах, состоящих в некоторых отношениях с рассматриваемым:

1) причинно-следственные связи, предполагающие информацию о синдромах, которые могут быть причиной данного синдрома или, наоборот, являться его следствием;

2) временные связи, позволяющие как прогнозировать состояние ребенка, так и восстанавливать возможный анамнез болезни;

3) ассоциативные связи, дающие возможность учитывать, на фоне каких состояний может развиваться данный синдром, фоном для каких синдромов он может служить и с какими синдромами может быть совместим, то есть какие синдромы могут встречаться одновременно;

- информацию о состояниях, взаимоисключающих друг друга;

- сведения о дифференцируемых синдромах (синдромах-конкурентах).

Все знания о синдромах подразделяются на декларативные - для описания самого синдрома (клиническая картина, дополнительные синдромы) - и процедурные, указывающие на то, как использовать знания в процессе диагностики.

Функциональная схема ДИН соответствует схеме классической ЭС: интерфейс, блок представления знаний, блок метазнаний, блок механизма логического вывода, блок рабочей области, блок пополнения и модернизации знаний.

2. Информационно-диагностическая система по наследственным болезням у детей «Диagen». Система «ДИАГЕН» - предназначена для консультативной помощи врачам при диагностике редко встречающихся и трудно распознаваемых заболеваний наследственной природы на долабораторном этапе обследования. База знаний системы включает сведения о 1200 моногенных и хромосомных болезнях, а модуль визуального представления данных - более 1000 фотографий больных с наследственными болезнями. Система включает три основных блока: диагностический, выдающий дифференциально-диагностический ряд; справочник, который предоставляет полную информацию о признаках и синдромах, хранящихся в системе (ориентирован в первую очередь на врача-педиатра широкого профиля); архив, который обеспечивает хранение и повторное использование данных о диагностированных больных.

Система может быть использована при решении вопроса о предварительном (долабораторном) диагнозе при подозрении на моногенную и хромосомную патологию как справочная система, а также в учебном процессе.

Система "ДИАГЕН" реализована для IBM-совместимых персональных компьютеров. Программное обеспечение написано на языке Borland C++ и функционирует под MS DOS.

3 Экспертная система Мутант.

Система Мутант обеспечивает проведение первичного скрининга заболеваний с целью получения диагностической информации, необходимой для организации более углубленного обследования пациентов. Предлагаемый подход к дифференциальной диагностике основан на интервьюировании пациента по определенной схеме. Вначале пациенту предлагается ответить на первичный (основной) опросник, включающий паспортные и некоторые антропометрические данные (масса тела, пульс, артериальное давление), личный и семейный анамнез, перечень основных жалоб, обычно приводящих пациента к врачу-терапевту. На основании компьютерной обработки ответов пациента делаются заключения двух типов: - о наличии риска определенных заболеваний (например, ишемической болезни сердца, сахарного диабета, хронического бронхита, рака легких и др.). В этом случае при отсутствии жалоб указывается, какие дополнительные исследования следует провести; - при наличии тех или иных жалоб пациент адресуется к соответствующему дополнительному (специализированному) опроснику, содержащему углубленную проработку каждого симптома по различным характеристикам, например, точная локализация болей, их характер, иррадиация, продолжительность, сопутствующие явления, условия прекращения и курирования. Для этого по каждому разделу предлагается набор альтернативных ответов. При обработке полученных ответов дополнительно используются такие элементы основного опросника как, например, сведения о поле, возрасте, анамнезе, факторах риска и других жалобах. На этом основании формируется диагностическое заключение в форме синдрома (например, сердечная или легочная недостаточность, синдром малоабсорбации и др.) или определенной нозологической единицы с указанием степени его вероятности в соответствии с количеством информационных синдромов, полученных от больного (50-75%, 76-95%, >95%). Врач должен учитывать, что диагностическое заключение носит рекомендательный характер. В дальнейшем врач может провести более углубленное обследование пациента, заведя в системе данные объективного осмотра пациента (осмотр головы, живота, ног и т.д.) и результаты дополнительных методов исследования (анализы крови, мочи, кала, мокроты и т.д.). В результате система уточняет раннее выставленные диагнозы.

Экспертная система Мутант состоит из следующих подсистем:

- подсистема консультаций (ПК);
- подсистема пополнения знаний (ППЗ).

Моделью представления знаний описываемой ЭС является продукционная система.

Фактологическими знаниями (базой фактов) ЭС Мутант являются:

- карта первичного опроса (КПО) и специализированные опросники (СО);
- список диагностируемых заболеваний;
- список анализов, рекомендуемых системой;
- список анализов, которые могут быть сделаны в данном медицинском учреждении;
- список врачей - специалистов, рекомендуемых системой;
- список врачей - специалистов, которыми располагает данное медицинское учреждение.

Процедурные знания (базы знаний) для системы содержат правила двух типов:

- правило 1-го типа - Если (условие), то (медицинское заключение);
- правило 2-го типа - Если (условие), то (список опросников). Здесь условием является список выбранных экспертом вопросов из карты первичного опроса и одного из специализированных опросников.

Медицинское заключение включает в себя:

- список предполагаемых диагнозов;
- список рекомендуемых системой анализов;
- список рекомендуемых системой врачей-специалистов.

Список опросников содержит номера специализированных опросников, которые необходимо предложить пациенту для уточнения медицинского заключения.

База знаний состоит из двух частей, в каждой из которых хранятся правила только одного типа. Каждое правило представлено одним термом. Все термы, соответствующие правилам, относящимся к одному опроснику, и группируются в одну цепь. Каждое правило при занесении в БЗ автоматически получает уникальный номер, который не может быть модифицирован, и удаляется из БЗ только при удалении этого правила.

Механизм вывода ЭС Мутант заключается в сопоставлении ответов пациента на вопросы карты первичного опроса и специализированных опросников с условной частью правил из БЗ. Поскольку задачей данной системы является выявление всех возможных предварительных заключений, то для ее решения применяется прямой вывод. При этом в зависимости от отношения количества ответов пациента, совпавших с условной частью правила 1-го типа, к общему количеству условий в данном правиле, система различает три степени достоверности диагнозов:

- диагноз маловероятен (50-75%);

- диагноз вероятен (76-95%);
- диагноз весьма вероятен (более 95%).

Для работы со знаниями врач-эксперт использует две подсистемы ведения БЗ, каждая из которых предназначена для работы со знаниями 1-го или, соответственно, 2-го типа.

Каждая из подсистем предлагает эксперту следующие виды работы:

- ввод нового правила в БЗ;
- коррекция правила в БЗ;
- удаление правила из БЗ;
- просмотр БЗ.

Применением базы знаний системы Мутант было ее совместное использование с базой данных медицинских карт. Данный подход позволяет не только хранить результаты сеанса работы системы с пациентом, но и дает возможность врачу занести в БД свои собственные наблюдения, результаты анализов и т.п.

В этих системах реализован интерактивный подход в использовании экспертной системы, когда в результате ответов пациента непосредственно формируется заключение. Однако авторы сочли возможным использование экспертной системы Мутант и в скрытом режиме в качестве контролирующей системы.

4. *Экспертная система FIRSTConsult.*

Ресурс FIRSTConsult (прежнее название - PDxMD) является частью MD Consult Clinical Knowledge System. Эта система научно обоснованной электронной клинической информации о первичной медицинской помощи предназначена для улучшения процесса принятия решений специалистами посредством предоставления специализированных диагностических средств и непрерывно обновляемых данных о последних тенденциях в области диагностики, вариантах ведения пациентов и результатах лечения пациентов. Она обеспечивает эффективное принятие решений, сочетая подход на основе передовой практики с гибкостью, необходимой для учета индивидуальных различий между пациентами и предпочтений врача; ускоряет клинические исследования с целью постановки диагноза и оптимизирует результаты, предлагая планирование научно обоснованного лечения и повышая эффективность использования времени и ресурсов; снижает риск ошибок и улучшает клиническую документацию, предоставляя всеобъемлющую и постоянно обновляемую информацию. Ресурс FIRSTConsult доступен в трех форматах: он-лайн, для карманных компьютеров и в печатном. Формат FIRSTConsult Online обновляется еженедельно и состоит из инструмента «Интерактивная дифференциальная ди-

агностика», научно обоснованных Файлов по медицинским состояниям, Файлов с информацией о пациентах, Справочных центров и Процедурных файлов. Формат FIRSTConsult Handheld включает в себя инструмент «Интерактивная дифференциальная диагностика» и Файлов по медицинским состояниям, обновляемых ежеквартально.

Файлы по медицинским состояниям охватывают более 450 медицинских состояний, которые чаще всего наблюдаются врачами первичной медицинской помощи. Каждое состояние включает в себя самую необходимую краткую информацию из других разделов; справочную информацию, включая коды ICD 9 и эпидемиологию; диагностику, включая дифференциальную диагностику, признаки и симптомы, предлагаемые вопросы для истории болезни, ссылки на литературу, полезную при принятии решений медицинскими учреждениями, и клинические тесты, которые можно провести; терапию с кратким перечнем терапевтических вариантов, подробными сведениями о лекарствах и проблемах пациентов и медицинских учреждений; результаты, включая эффективность терапии, прогноз и осложнения; профилактику с указанием факторов риска, рекомендаций в отношении образа жизни и сохранения здоровья, и скрининг; ресурсы, включая основную литературу и ссылку на информацию для пациента; а также научные данные, увязанные с библиографическими ссылками и клиническими руководствами.

Файлы по дифференциальной диагностике позволяют пользователям ознакомиться с таблицей диагностических алгоритмов, в которой перечисляются потенциальные диагнозы с разбивкой по возрасту и распространенности для более чем 330 признаков и симптомов. Потенциальные критические состояния показаны красным. Выбор диагноза выводит на экран краткую сводку, включая клиническую картину начала заболевания, соотношение процентов среди мужчин и женщин, особенности у разных этносов, характер, географические особенности, клинический картину течения заболевания и сопутствующие заболевания со ссылкой на файлы по медицинским состояниям. Файлы с информацией о пациентах содержат раздаточные материалы для пациентов, написанные в формате вопросов и ответов. Эти материалы имеют ссылку на файлы по медицинскому состоянию и могут быть адаптированы под конкретного пациента. Раздаточные материалы доступны различных языках в различных форматах и предназначаются для пациента, у которого было диагностировано данное состояние, либо для общего сведения.

Рассмотрим **обобщенный алгоритм работы** диагностической экспертной системы.

В алгоритме учитывается, что система медицинской диагностики должна позволять на основе неполных данных ставить диагноз с некоторой точностью, и выдавать, насколько точно она определила болезнь.

Входные данные можно разбить на два больших блока. Первый блок - это данные, поступающие из пользовательского интерфейса. Второй блок - это содержимое базы знаний, заполненной экспертом. База знаний должна храниться на жестком диске в виде четырех файлов. Структура базы знаний будет описана более подробно в одном из последующих пунктах.

Таким образом, входными данными для разработки алгоритма программы системы медицинской диагностики являются:

- ответы пользователя на вопросы системы;
- база данных с описаниями симптомов;
- база данных с описанием болезней;
- таблица соответствий между болезнями и симптомами;
- таблица «весов» (вероятностей) симптомов для болезней;

Выходными данными экспертной системы является диагноз, построенный на основе наблюдаемых симптомов и базы знаний о болезнях. Этот диагноз появляется на экран дисплея ПВМ.

В процессе работы система предлагает несколько сгенерированных рабочих версий окончательного диагноза, и по требованию пользователя «отсеивает» лишние гипотезы, которые имеют вес, меньший, чем некоторое значение, заранее заданное разработчиком программы или опытным пользователем.

Например, если в процессе работы сформировалось 4 версии с вероятностями от 71 до 93 %: диагноз №1 – 93%; диагноз № 2 – 86%; диагноз № 3 – 79%; диагноз № 4 – 64% и порог уверенности, заданный разработчиком программы – 80%, то система показывает все версии, вероятности которых больше или равны 80% (например, их 2 - Диагноз №1 и Диагноз №2 из сформированных).

При заполнении базы знаний необходимо привлекать эксперта.

База знаний представляет собой совокупность следующих баз данных:

- база данных с описаниями диагнозов.
- база данных с описаниями симптомов.
- таблица соответствий симптомов и диагнозов.
- таблица весов симптомов для диагнозов.

Вес – это относительная величина значимости каждого симптома для конкретного диагноза.

Кроме того, в экспертной системе предусмотрен порог уверенности – число, заранее определенное пользователем, означающее максимальный вес диагноза в данном случае, ниже которого гипотезы не рассматриваются.

Например, структуры данных представленные в Paradox 7.0 показаны в таблицах 1–2.

Работу с этими таблицами обеспечивают DBE и DBD, поставляемые в комплекте с Borland Delphi для работы с базами данных распространенных форматов.

Таблица 1 – Таблица диагнозов

	Наименование поля	Тип	Длина	Краткое описание
	Num	numeric	11.	Идентификационный номер диагноза
	Name	character	16. 5	Наименование диагноза
	FileName	character	21. 2	Имя файла с рекомендациями

Таблица 2 – Таблица симптомов

	Наименование поля	Тип	Длина	Краткое описание
	Num	numeric		Идентификационный номер
	NameS	character	15	Наименование симптома
	FileName	character	12	Имя файла с рекомендациями

Все значения полей таблицы соответствий имеют тип `numeric`. Число в пересечении колонок является «указателем» на тот диагноз, на который указывает определенный симптом, либо на тот симптом, на который указывает определенный диагноз. Всем значениям полей таблицы весов назначается тип `numeric`. Таблица имеет размерность $[n,m]$, где n – число диагнозов, m – число симптомов. Число в пересечении колонок является весом данного симптома для данного диагноза.

Работа программы начинается с функции инициализации, которая ответственна за присвоение данным начального значения. После чего управление получает процедура «Главного меню». Далее, в зависимости от действий пользователя, происходит вызов одной из ниже перечисленных процедур:

- работа с базами знаний;
- непосредственный запуск экспертной системы;
- настройки;
- контекстно-зависимая справочная система.

На каждом шаге алгоритма у системы предусмотрен «откат», т.е. пользователю предоставлена возможность вернуться в предыдущее состояние системы, нажав на программную кнопку «Назад». На вход системы изначально поступает неполная информация в виде данных, введенных пользователем с клавиатуры. Вследствие чего система не может однозначно поставить диагноз.

Для решения проблемы выбора диагноза используется следующий алгоритм:

1. Сбор предварительной информации. При входе в систему пользователю предоставлен список всех симптомов, имеющихся в базе. Напротив выборочных пунктов списка симптомов пользователь должен выставлять «галочки» отмечая симптомы, которые он наблюдает. При нажатии пользователем программной кнопки «Далее» система переходит к следующему пункту.

2. Составление первоначального списка диагнозов, к которым подходит данный набор симптомов. После ввода пользователем симптомов, система анализирует, к каким диагнозам принадлежат данные симптомы по следующему алгоритму:

- а) создается список пустой первоначальный список диагнозов
- б) открывается таблица соответствий;
- в) для каждого i -го симптома, $i=1..m$, где m –максимальное число симптомов по таблице соответствий циклом делаем полный перебор всех эле-

ментов (диагнозов), стоящих в столбце данного симптома и определяем предварительный диагноз;

г) если диагноз уже имеется в списке диагнозов, то переходим к пункту е);

д) если элемент таблицы равен 0, то диагноз в список диагнозов не включается;

е) переходим к проверке следующего симптома.

ж) если все симптомы (i) перебраны, то первоначальный список болезней сформирован;

з) закрываем таблицу соответствий.

Каждый симптом может принадлежать сразу нескольким диагнозам, поэтому в первоначальный список диагнозов следует уточнить, используя таблицу весов. И следующим пунктом обобщенного алгоритма будет уточнение полученных данных.

3. Уточнение первоначального списка диагнозов.

Имея начальный список диагнозов, система осуществляет их дифференциацию. Самыми распространенными методами логического вывода являются прямая цепочка рассуждений (прямой вывод) и обратная цепочка рассуждений (обратный вывод). В основном, при решении задач диагностики используется обратный вывод. Можно сказать, что обратный вывод более эффективен, когда пользователь должен выбирать из набора возможных последствий в случае медицинской диагностики. В предлагаемой системе реализуется механизм смешанного вывода, который позволяет и прямой вывод от фактов к заключениям, и обратный – чтобы подтвердить или опровергнуть гипотезу. В процессе уточнения информации система, задавая пользователю вопросы, осуществляет «отсеивание» лишних гипотез, имеющих малый вес. Для просчета веса гипотез система должна открыть таблицу весов. В таблице весов указывается вес данного симптома для данного диагноза на основе априорных знаний эксперта. Порог уверенности заранее задается в настройках приложения и использованием интуитивно понятного интерфейса.

4. Принятие окончательного решения.

В процессе предыдущих шагов выявляется несколько версий окончательного результата, которые система должна распределять по порядку возрастания вероятности того или иного диагноза. Вероятности диагнозов считаются по таблице весов. Предлагается следующий алгоритм подсчитывания веса:

а) выбирается диагноз из списка диагнозов, сформированного на предыдущих этапах;

- б) система просматривает, какие симптомы из списка симптомов имеют отношение к данному диагнозу;
- в) суммируются веса, определенные по таблице весов, всех симптомов, имеющих отношение к данному диагнозу;
- г) фиксирование конечного веса диагноза;
- д) после подсчета весов всех диагнозов выбирается диагноз, имеющий максимальный вес;
- е) система выбирает те диагнозы, оценки вероятности которых выше «порога».

Для конечного подсчета вероятности предлагается суммирование всех элементов таблицы весов имеющих отношение к болезни и нормировка всех конечных вероятностей с целью «укладывания» их в промежуток от 1 до 100 с целью получения окончательного ответа в процентах.

5. Представление конечного результата.

Система предоставляет врачу диагнозы, которые были выбраны в предыдущем пункте в порядке процентного убывания на экран монитора ПЧМ и предоставлять пользователю возможность корректировать базу знаний на основе новых экспертных знаний и своего опыта.

На основании рассмотренного обобщенного алгоритма разработана экспертная система дифференциальной диагностики внутренних болезней.

Базовые скрин-шоты, представляющие интерфейс данной системы, представлены на рисунках 1-5.

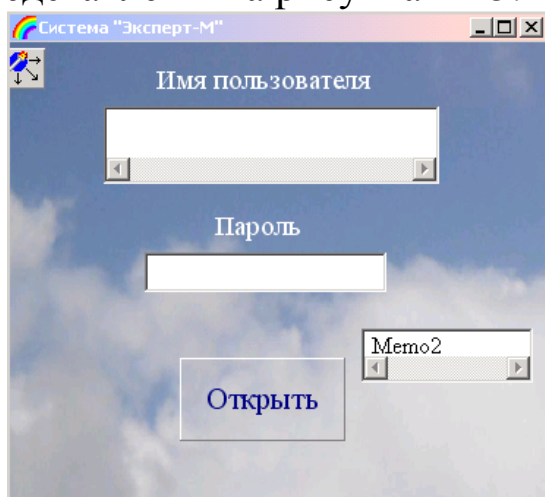


Рисунок 1 – Форма окна входа в систему

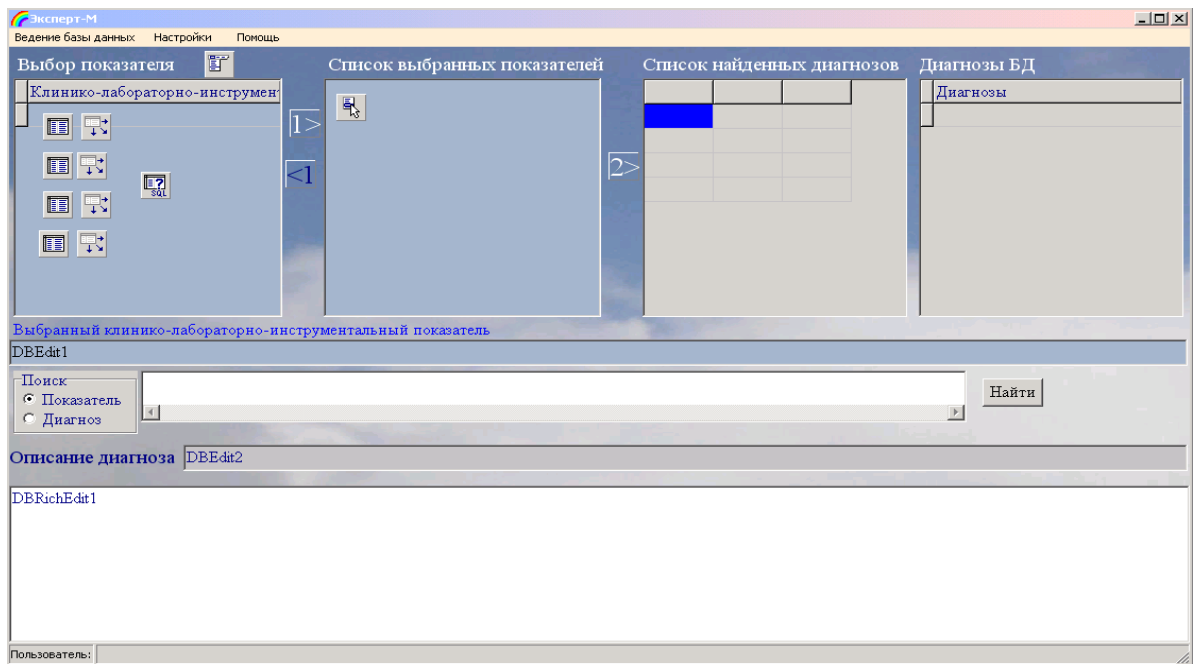


Рисунок 2 – Форма основного окна программы

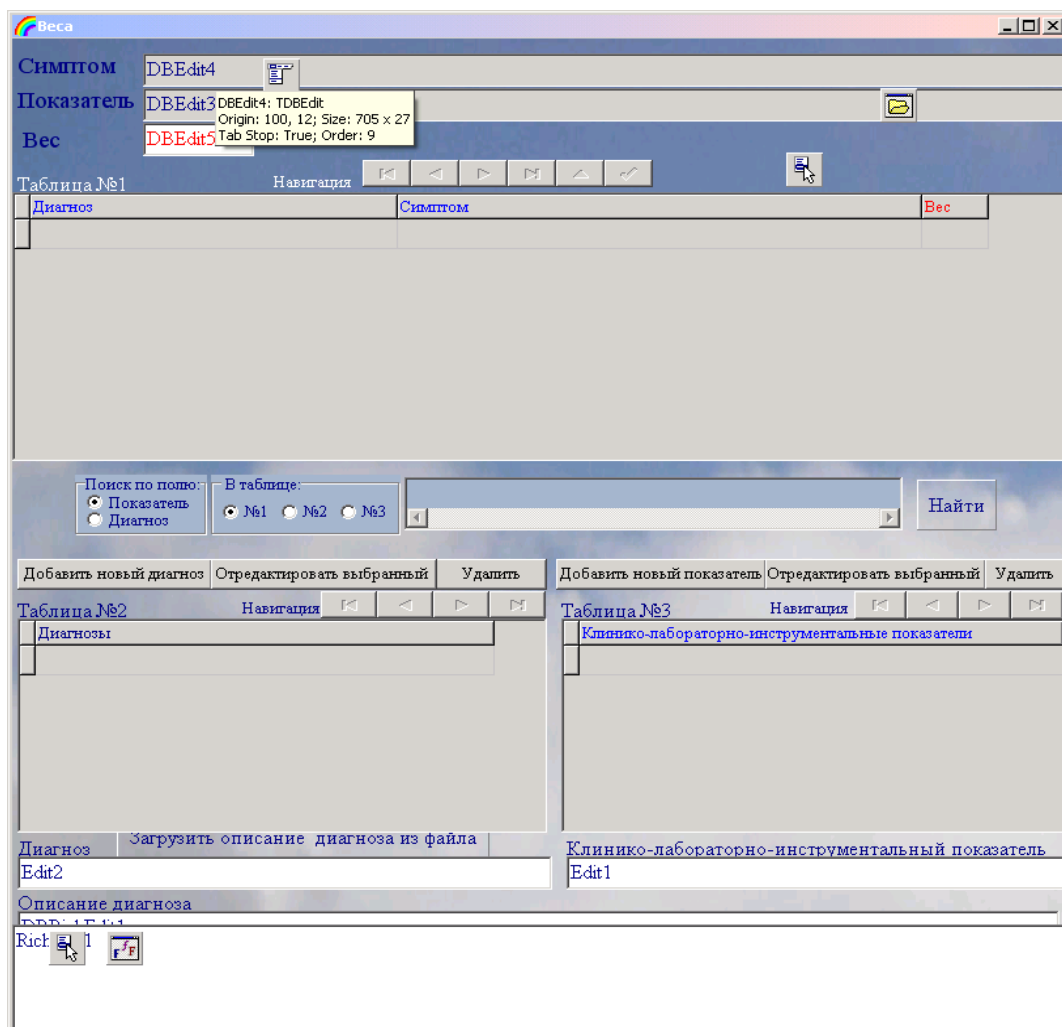


Рисунок 3 – Форма окна редактирования базы данных

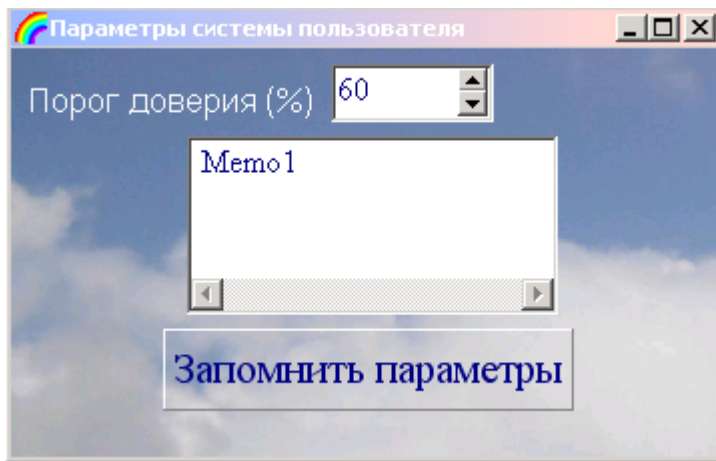


Рисунок 4 – Форма окна редактирования параметров системы

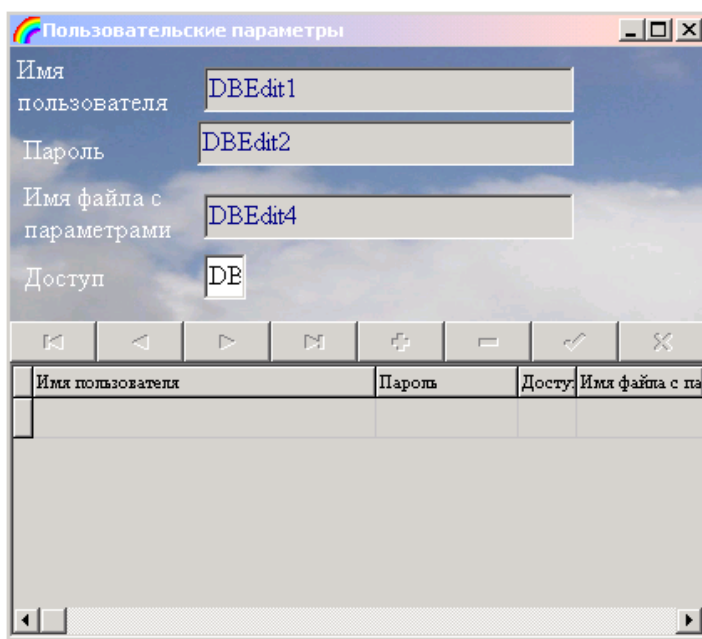


Рисунок 5 – Форма окна редактирования параметров пользователей

База данных состоит из четырех таблиц и разработана по схеме представленной на рисунке 6.

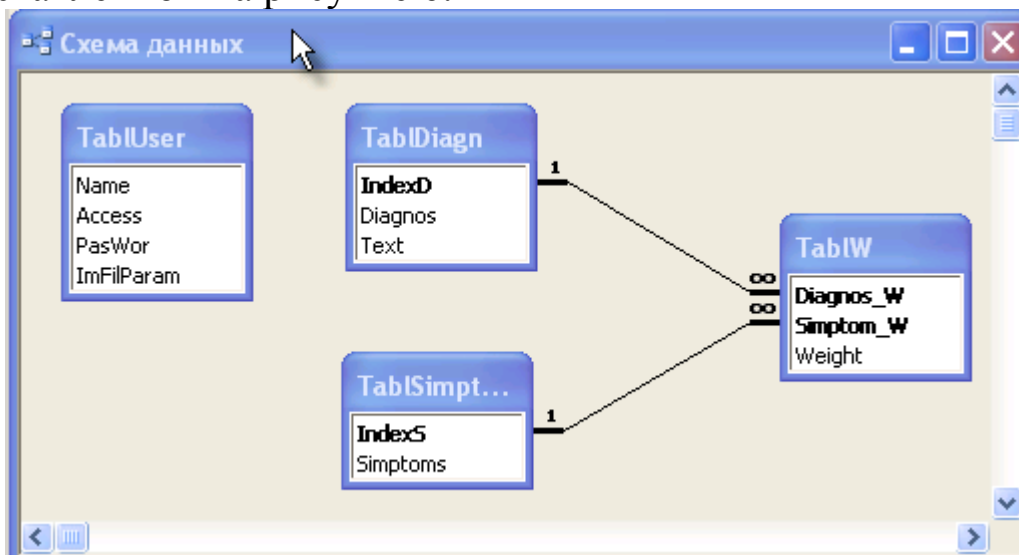


Рисунок 6 – Схема данных

Заполнение базы данных производится экспертом в разработанном приложении. Заполнению подлежат таблицы диагнозов, таблицы клинико-лабораторно-инструментальных показателей и расстановка соответствующих весов в таблице весов. Пример заполнения представлен на рисунке 7.

The screenshot shows a software application window titled "веса". At the top, there is a form with the following fields:

- Симптом**: боль в подложечной области: тупая
- Показатель**: Острый холецистит
- Вес**: 0

Below this is a table labeled "Таблица №1" with columns "Диагноз" and "Вес".

Диагноз	Симптом	Вес
Острый холецистит	анализ мочи: пиурия	0
Острый холецистит	болезненная, диффузная припухлость	0
Острый холецистит	болезненность при пальпации	3
Острый холецистит	боли в животе: в верхней части с иррадиацией влево	0
Острый холецистит	боль в животе: спастический характер	20
Острый холецистит	боль в животе: строго локализованная, связанная с приемом:	0
Острый холецистит	боль в животе: умеренная боль в правой подвздошной обл	0
Острый холецистит	боль в подложечной области: тупая	0

Below the table is a search section with "Поиск по полю:" (Показатель, Диагноз) and "В таблице:" (№1, №2, №3) options, and a "Найти" button.

The bottom section contains two detailed tables:

- Таблица №2** (Диагнозы):
 - Инфаркт миокарда
 - Остеоартроз
 - Острый аппендицит** (highlighted)
 - Острый панкреатит
 - Острый холецистит
 - Пиелонефрит
- Таблица №3** (Клинико-лабораторно-инструментальные показатели):
 - снижение аппетита
 - Боль в грудной клетке: продолжительность > 20-30
 - Боль в суставе: спонтанные боли, усиливающиеся п
 - УЗИ: внутрисосудистое исследование выявило обс:
 - УЗИ: лоцируются конкременты в чашечно-лоханочн
 - УЗИ: признаки гидронефроза

Below these tables is a "Диагноз" section with "Острый аппендицит" and its description: "Острый аппендицит проявляется умеренной болью, которая начинается либо непосредственно в правой подвздошной области, либо дебютирует в эпигастрии и перемещается в правую подвздошную область спустя 2-3 часа от начала заболевания".

Рисунок 7 - Пример заполнения базы данных

Пример работы с базой данных представлен на рисунке 8.

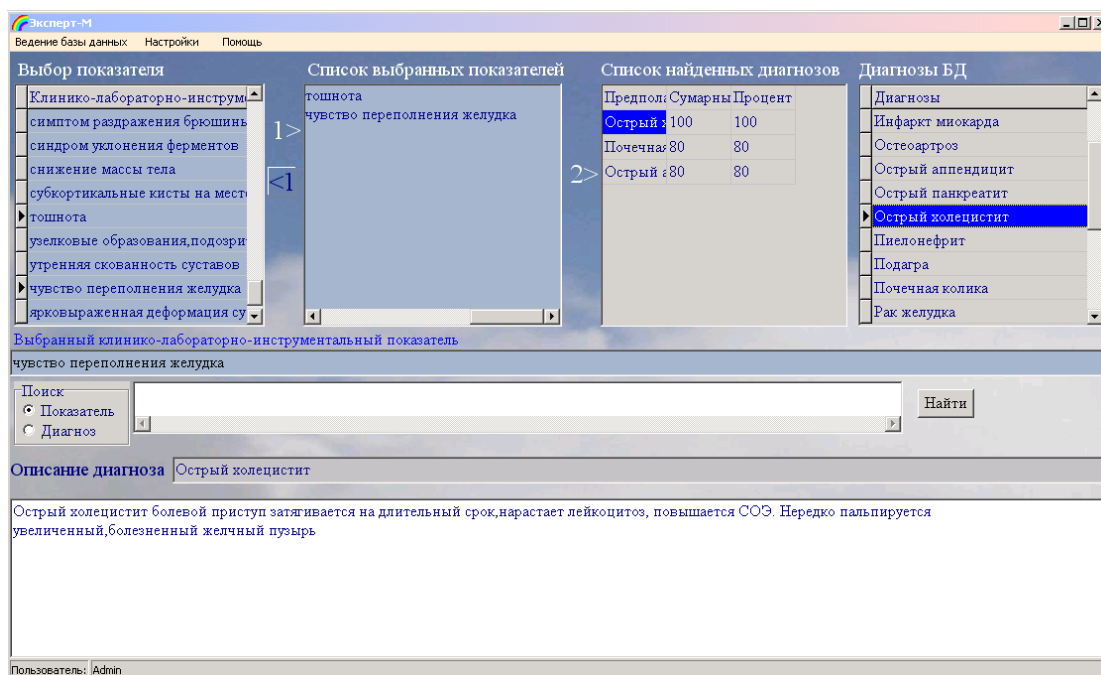


Рисунок 8 - Пример работы с базой данных.

Обобщенная структура экспертной системы представлена на рисунке 9.

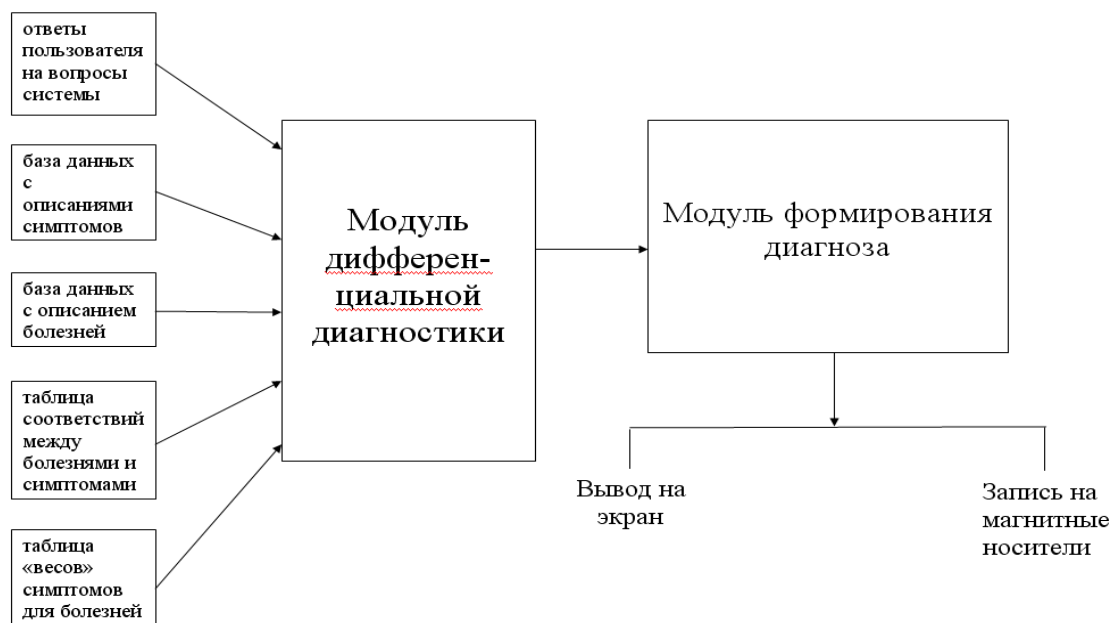


Рисунок 9 - Обобщенная структура экспертной системы

В модуль дифференциальной диагностики поступают данные из комплекса программ, реализующих диалог пользователя с ЭС как на стадии ввода информации, так и получения результатов (интерфейс пользователя) а также данные из базы данных с описаниями симптомов и болезней и данные из таблиц весов симптомов и болезней и соответствия меж-

ду болезнями и симптомами. Модуль дифференциальной диагностики - программа, моделирующая ход рассуждений эксперта на основании знаний, имеющихся в БЗ.

Итогом работы модуля дифференциальной диагностики является совокупность возможных диагнозов, соответствующих всем выбранным пользователем симптомам.

Далее в модуле формирования диагноза происходит анализ и отсеивание диагнозов суммарный вес которых, ниже порога чувствительности.

Полученные результаты выводятся на экран или записываются на магнитные носители.

Инструкция пользователя по эксплуатации экспертной подсистемы для дифференциальной диагностики внутренних болезней.

При запуске приложения открывается окно, представленное на рисунке 10.

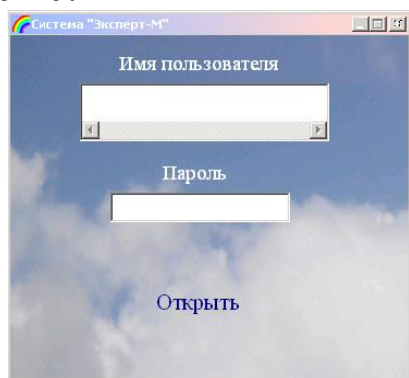


Рисунок 10 – Окно входа в систему

При вводе имени пользователя и пароля открывается основное рабочее окно программы, представленное на рисунке 11.

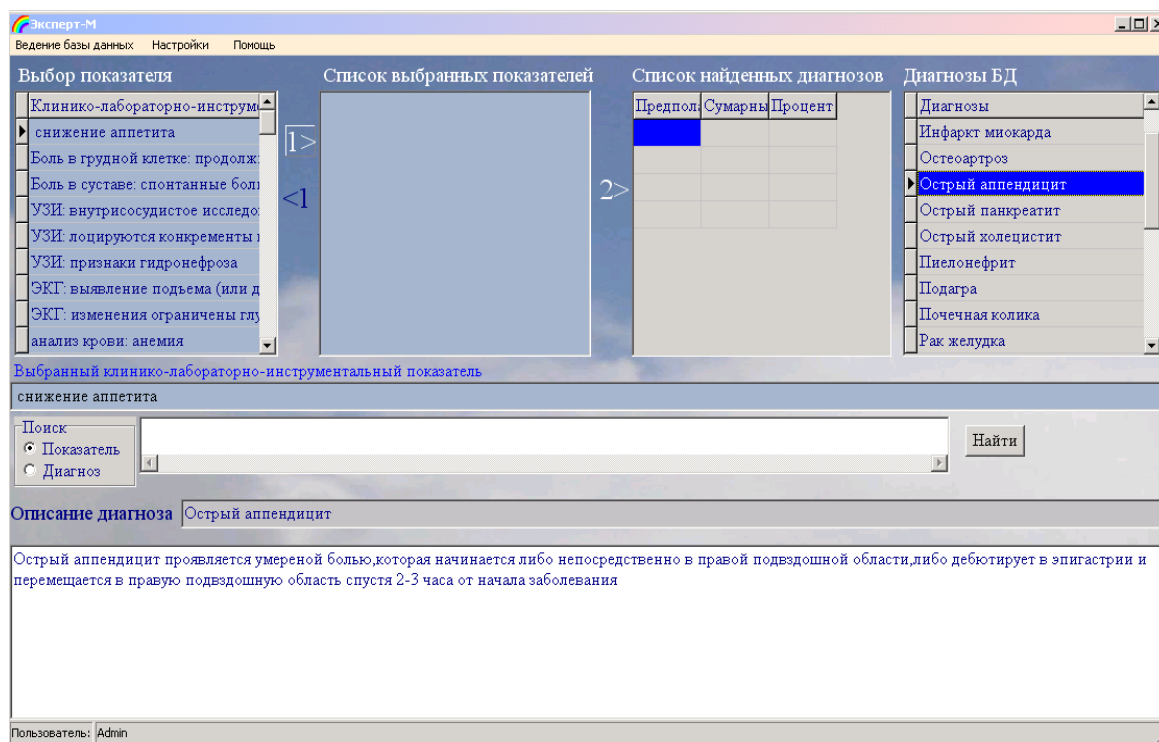


Рисунок 11 – Рабочее окно программы

В начале необходимо выбрать клиничко-лабораторно-инструментальный показатель таблицы «Выбор показателя» и нажать программную кнопку «1>». При этом выбранный показатель отобразится в списке выбранных показателей. Аналогичным образом следует выбрать все необходимые показатели, соответствующие собираемому анамнезу. Далее следует нажать программную кнопку «2>». При этом в списке найденных диагнозов отобразятся предполагаемые диагнозы, суммарный вес показателей в диагнозе и процентное соотношение суммарных весов показателей к максимальному суммарному показателю. Описание полученных предполагаемых диагнозов можно просмотреть при нажатии левой кнопки мыши при выделенном выбранном диагнозе таблицы «Диагнозы БД». Для удобства пользователя организован поиск выбранных диагнозов или показателей по базе данных. Для поиска необходимо ввести в поисковое окно искомый текст, пометить в окне поиска категорию поиска – диагноз/ показатель, и нажать программную кнопку «Найти». При этом найденная текстовая информация отразится в таблице диагнозов или симптомов в зависимости от категории поиска. Пользователь может изменить порог чувствительности, выбрав подпункт «Параметры системы» пункта меню «Настройки». При этом должно появиться окно «Параметры системы», представленное на рисунке 12. В открывшемся окне пользователь может изменить порог чувствительности, и нажать программную кнопку «Запомнить параметры».

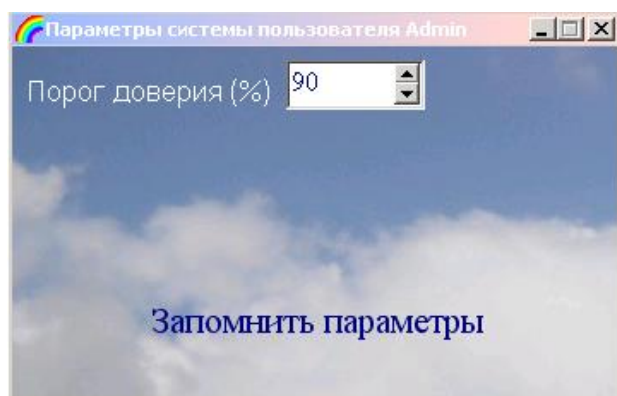


Рисунок 12 – Окно редактирования параметров системы

Для ведения базы данных необходимо входить в систему под именем администратора с паролем администратора и доступом администратора (полный доступ). В этом случае будут доступны пункты меню «Ведение база данных» и «Пользователи». Администратор при необходимости может добавить нового пользователя для работы с программой. Для этого необходимо выбрать подпункт «Пользователи» пункта меню «Настройки». При этом в открывшемся окне, представленном на рисунке 13, администратор может ввести в таблицу пользователей имя нового пользователя, пароль, доступ (полный доступ – 1, ограниченный - 0) и имя файла настроек очередного пользователя.

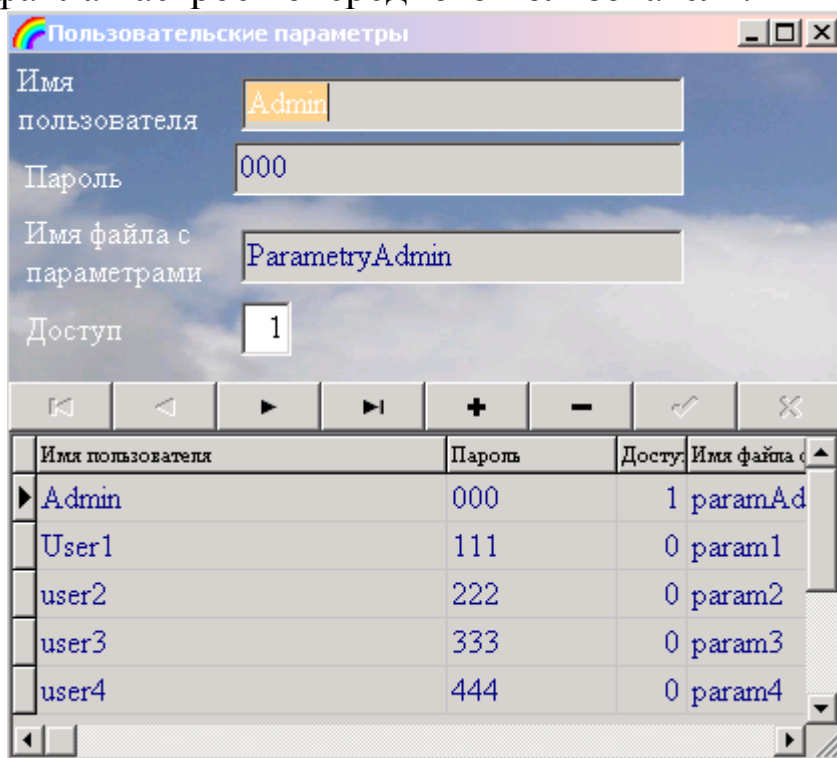


Рисунок 13 – Окно редактирования параметров пользователей

Обновлять базу данных рекомендуется эксперту. Для этого эксперт должен войти в программу под именем администратора с паролем администратора и доступом администратора. Далее следует выбрать пункт меню «Ведение баз данных», после чего откроется окно, представленное на рисунке 14.

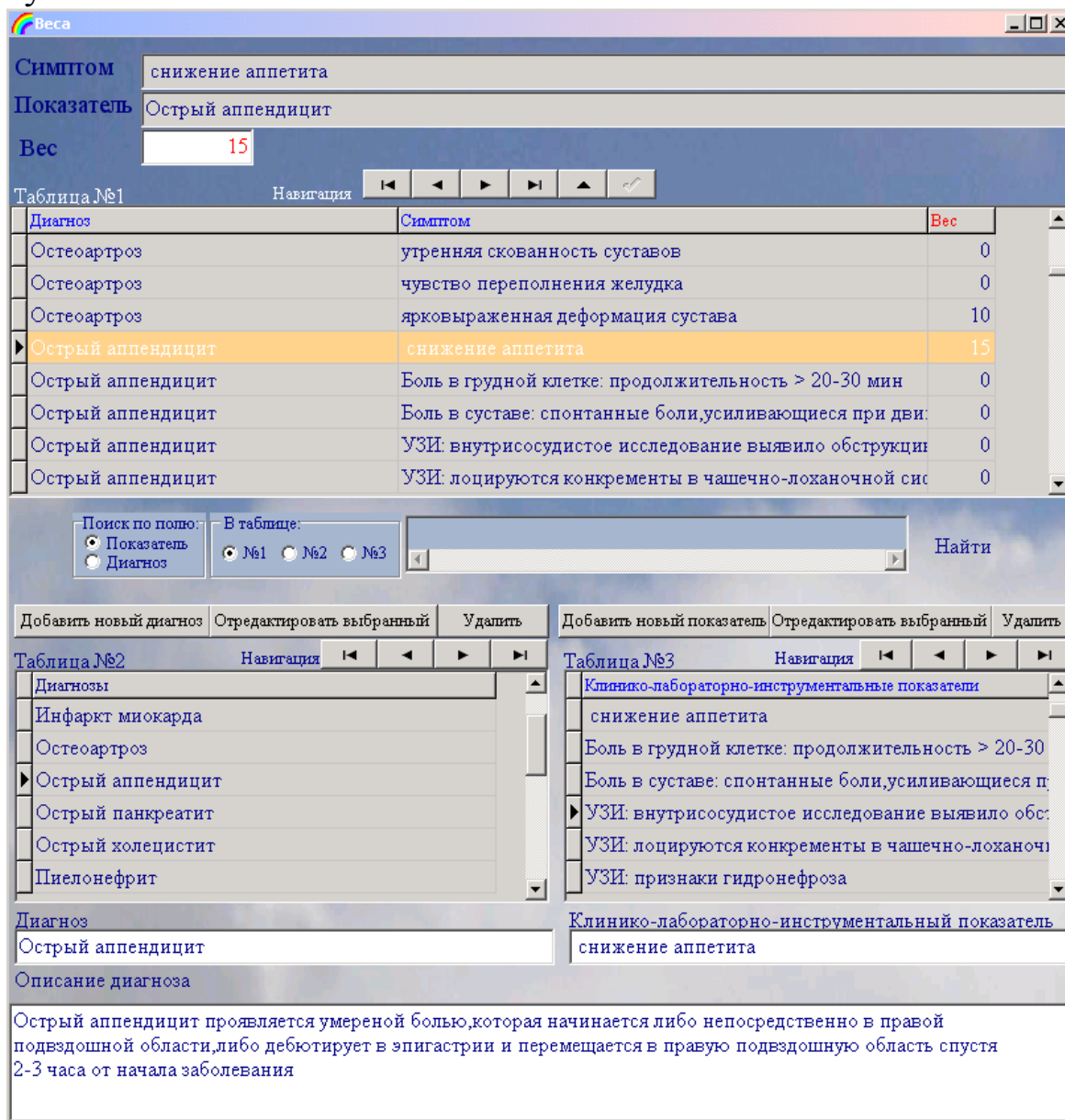


Рисунок 14 – Окно редактирования базы данных

Описываемую болезнь следует ввести в поле, озаглавленное «Диагноз», после чего нажать программную кнопку «Добавить новый диагноз». Введенный диагноз при этом отобразится в таблице №1 и таблице №2. Причем следует отметить, что в таблице №1 новый диагноз отобразится не один раз, а в комбинации со всеми клинико-лабораторными показателями, введенными ранее, для возможности последующей расстановки весов. Чтобы отредактировать диагноз, нужно выбрать его в табли-

це № 2 путем нажатия левой кнопки мыши, произвести необходимые изменения в поле «Диагноз» и нажать программную кнопку «Отредактировать выбранный». Для удаления введенного ранее диагноза, его следует выбрать в таблице №2 и нажать программную кнопку «Удалить». Для краткого описания болезни, она выбирается в таблице № 2, описание вводится в поле «Описание диагноза», после чего следует нажать программную кнопку «Отредактировать выбранный». Для введения симптомов болезней, а также результатов лабораторных исследований пациента следует использовать поле «Клинико-лабораторно-инструментальный показатель», после чего нажать кнопку «Добавить новый показатель». При этом симптом отобразится в таблице №1 в комбинации с ранее введенными диагнозами и в таблице №3. Редактирование и удаление симптомов производится аналогично диагнозам. Для расстановки весов в таблице №1 нажатием левой кнопки мыши выбирается строка с нужной болезнью и симптомом, далее в поле «Вес» следует ввести число, соответствующее, по мнению специалиста, влиянию данного симптома на постановку рассматриваемого диагноза.

В экспертной подсистеме для дифференциальной диагностики внутренних болезней предусмотрена справочная система. Для ее вызова следует выбрать пункт меню «Помощь», после чего откроется окно, представленное на рисунке 15.

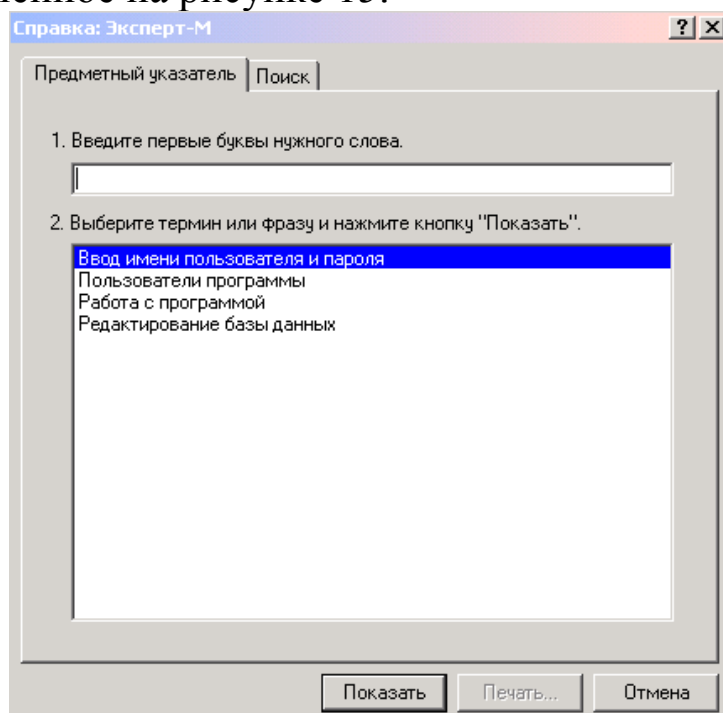


Рисунок 15 – Справочное окно

В открывшемся окне можно либо ввести искомый термин в поле под номером 1, либо выбрать из предложенных в поле под номером 2. После

этого следует нажать программную кнопку «Показать». Например, при выборе пункта «Работа с программой» откроется справка, представленная на рисунке 16.

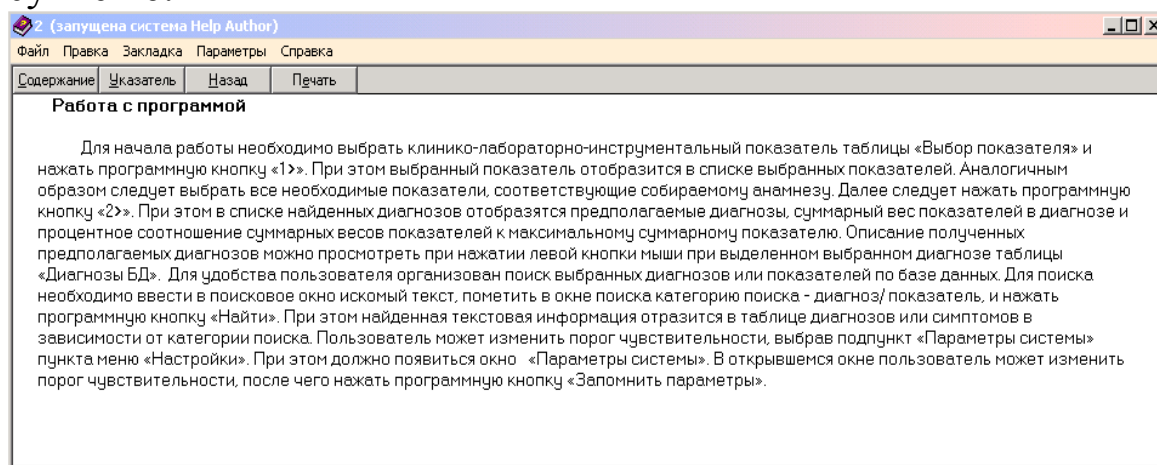


Рисунок 16 – Пример справочной информации.

Порядок выполнения работы:

1. Изучить теоретические материал.
2. Под руководством преподавателя выбрать предметную область дифференциальной диагностики внутренних болезней.
3. Запустить оболочку экспертной системы дифференциальной диагностики внутренних болезней и экспериментальным путем освоить интерфейс системы.
4. Заполнить оболочку согласно выбранной предметной области.
5. Оценить качество созданной системы дифференциальной диагностики.
6. Оформить отчет (включая скрин-шоты интерфейсной части).

Контрольные вопросы:

1. В чем заключается функциональное назначение дифференциальной диагностики?
2. В чем заключаются основные принципы дифференциальной диагностики?
3. Перечислите наиболее используемые в настоящее время в России и за рубежом экспертные системы дифференциальной диагностики.
4. Какое влияние оказывают полисиндромальные процессы на формирование диагноза?
5. Как проверяется качество работы диагностической системы?
6. Что входит в типовой состав экспертной системы дифференциальной диагностики?

7. Какие функциональные модули (и в какой последовательности) входят в состав обобщенного алгоритма работы экспертной системы дифференциальной диагностики?

ЛАБОРАТОРНАЯ РАБОТА №5: ПОСТРОЕНИЕ РЕШАЮЩИХ ПРАВИЛ

Цель работы: овладение навыками структурно-параметрической идентификации диагностических правил с применением средств вычислительной техники.

Краткие теоретические сведения.

Под диагностическими правилами понимается процедура вывода заключения о соотношении состояния анализируемого объекта или процесса к определенному классу или области на основании временно-пространственной регистрации существенных характеристик.

Любой объект (процесс) с точки зрения диагностики подвергается анализу со стороны исследователя, который, как правило, априори знает, какие существенные характеристики ему следует регистрировать для решения диагностической задачи. То есть, в этом случае, исследователь уже владеет набором диагностических правил, которые либо опровергают, либо подтверждают выдвинутую им рабочую гипотезу о состоянии объекта. Так как о каждом состоянии объекта может выдвигаться различное количество гипотез, то, следовательно, диагностические правила каждой из них не должны в случае объединения поглощать друг друга, и, вообще говоря, должны иметь минимальное количество пересечений как по регистрируемым параметрам, так и по диапазонам их изменений.

В общем случае диагностическое правило имеет вид, например, продукции: если значение $P=P_0$, то состояние $S=S_0$.

$$P=F(S, t, dS), \quad (1)$$

где S - состояние; t - время; dS - диагноз изменения характеристик состояния.

Если зависимость (1) достаточно хорошо идентифицирована (с заданной степенью точности или неопределенности), то нетрудно построить эксперто-диагностическую систему продукционного типа с указанием исследователю технологии реализации необходимой информации для достаточно достоверной диагностики гипотетического состояния.

Рассмотрим **логический механизм** синтеза правила (1).

1 этап. Организация мониторинга состояния заданной глубины и полноты.

2 этап. Выделение множества ортогональных и информативных признаков с точки зрения вариативности. То есть, с одной стороны, селективируем сильно коррелированные характеристики, с другой стороны, отби-

раем те из них, вариативность которых (отношение дисперсии к среднему значению) выше определенного порогового уровня (например 10%).

3 этап. Кодирование состояний (лучше в двоичном коде): с учителем - то есть исследователь знает состояния, без учителя - выполняется кластер-анализ и задаются состояния или вводится пороговый принцип. Таким образом, получаем значения «логической» функции $Y=(Y_{i1}, Y_{i2}, Y_{i1})$. Если состояний не много, то рекомендуется применять унитарное кодирование с минимизацией Хемингового расстояния соседних состояний.

4 этап. Кодирование значения признакового пространства, следующим образом (во всех случаях рекомендуется унитарный код). По каждому оставленному признаку выделяем определенный набор состояний, как попадание значения признака в определенный диапазон. Диапазон определяется либо:

1) Экспертом, исходя из его знаний и жизненного опыта.

2) Исследователем, по анализу частоты распределений значений и личного опыта. При достаточно небольшом количестве признаков анализ гистограммы рекомендуется проводить визуально, наблюдая все признаки одновременно (в концепции системный подход).

3) Автоматически (с применением ЭВМ) по следующему алгоритму.

Исследователь задает количество состояний по каждому признаку n_i (каждое из них кодируется, желательно в унитарном коде). Определяется медиана M_0 и дисперсия G_0 . Определяется удельное отклонение как $G_y = G_0/(n_{i-1})$. В качестве первого диапазона (состояния) выбирается величина внутри диапазона $M_0 \pm G_0$. Все значения X_i попавшие в данный диапазон кодируются определенным состоянием S_0 . Величина n_i декрементируется и повторяется описанный процесс над «оставшимися» данными. Так продолжается до тех пор, пока n_i не станет равно 0 и всем оставшимся значениям будет присвоено состояние S_n . Граничные значения $M_0 \pm G_{y0}$ либо включаются в одно из состояний, либо, что более оптимально, кодируются знаком переходной функции.

5 этап. Определяем функциональные зависимости между полученными булевыми функциями (парные и множественные) и парное Хеминговое расстояние. Те признаки, у которых это расстояние равно нулю, селекционируются путем оставления одного из них с наибольшей вариативностью.

Явный вид логической зависимости между булевыми переменными $X_k, k=1, m$ определяются следующим образом. На первом шаге проверяются условия независимости: поскольку каждая булева функция может иметь два значения истинности, то m булевых функций может образовывать 2^m комбинаций значений истинности. Согласно определению

m-булевых функций независимы, если в совокупности при всех возможных значениях аргументов они могут принимать 2^m комбинаций значений истинности. Т.е для проверки независимости необходимо вычислить их изображающие числа и проверить, образуют ли они полный набор чисел. Если да, то функции независимы, в противном случае - зависимы.

На втором шаге в базисе булевых функций выписывают в последовательные строки изображающие числа и определяют какие числа отсутствуют в наборе столбцов (повторяющиеся значения чисел считают один раз). Столбцы набора представляют собой комбинации значений истинности функций X_1, \dots, X_m , при которых соответствующие элементарные произведения составленные из X_1, \dots, X_m истинны.

Таким образом, если идентифицируется зависимость:

$$F(X_1, \dots, X_n) = 1 \quad (3),$$

то, следовательно, имеющиеся в наборе столбцы указывают номера тех колонок базиса в (X_1, \dots, X_n) , которые совпадают с номерами изображающего числа $\#F(X_1, \dots, X_m)$, на которых функция F истинна.

Например, пусть задан протокол мониторинга трех логических функций:

$$\begin{array}{l} X_1 \quad 11001010 \\ X_2 \quad 10101100 \\ X_3 \quad 11001100 \end{array}$$

Выпишем последовательно все столбцы в этом наборе изображающих чисел как строки и укажем справа их десятичные значения:

$$111=7, 101=5, 010=2, 000=0, 111=7, 110=6, 001=1, 000=0$$

Видно, что десятичные эквиваленты 3 и 4 отсутствуют, а это означает, что по отношению и в (X_1, X_2, X_3) изображающее число связи $F(X_1, X_2, X_3) = 1$ имеет вид $\#F(X_1, X_2, X_3) = 1$.

Минимизируя полученную функцию, получаем:

$$\#F = \bar{X}_1 \bar{X}_3 + X_2 X_3 + X_1 \bar{X}_2 = 1$$

Проверяем:

X1	X2	\bar{X}_3	$\bar{X}_1 X_3$	$X_2 X_3$	$\bar{X}_1 \bar{X}_2$	F
1	1	1	0	1	0	1
1	0	1	0	0	1	1
0	1	0	1	0	0	1
0	0	0	1	0	0	1
1	1	1	0	1	0	1
0	1	1	0	1	0	1

1	0	0	0	0	1	1
0	0	0	1	0	0	1

Таким образом, определяется как логические функции связаны между собой.

6 этап. Идентифицируем логические функции $Y=F(X)$ - парная зависимость и/или $Y=F(\{X\})$ (4) - множественная зависимость. Заметим, что возможен вариант отсутствия тех или иных функциональных зависимостей.

7 этап. Переходим от полученных булевских функций либо к продукционным диагностическим правилам, либо к схмотехническому решению идентификационного диагностического устройства. Однако, второй вариант менее устойчив и мобилен в случае достаточно быстрого изменения окружающей среды, приводящего к изменению в функционировании анализируемого объекта (системы, процесса), а, следовательно, и вида идентифицированных функций.

Как и во множественном регрессионном анализе, при синтезе зависимостей (4) для получения более строгого результата (минимизации пересечений понятий в диагностических, классификационных правилах каждого состояния) рекомендуется руководствоваться правилом максимальной организации (независимости) факторного пространства. Для этого необходимо добиться максимальной независимости X между собой, т.е. в идеале не должно существовать функциональных зависимостей между X_i . Т.е., если на пятом этапе идентифицируется $F(x)=1$ (3), то необходимо изменить множество X : либо путем исключения переменных (по критерию вариативности), что чревато в общем случае, потерей информации; либо изменить кодирование вводимых сигналов путем уменьшения количества состояний и/или изменения (экспертным путем) диагностических классов состояний. При достаточно мощной вычислительной технике и сравнительно небольшом размере факторного пространства (до 100 признаков) эти проблемы могут быть решены переборным путем. В противном случае, следует применять методы целенаправленного случайного поиска.

Как и в регрессионном анализе возможно формирование продукционных диагностических правил с учетом фактора запаздывания.

Порядок выполнения работы.

1. Самостоятельно изучите теоретический материал.

2. Согласно номеру варианта (равен порядковому номеру студента в журнале группы) N сформируйте протокол мониторинга наблюдения за процессом X , следующим образом:

Индекс переменной X	Индекс протокола Z (см. таблицу 1)
1	$\text{mod}(n,8)+1$
2	$\text{mod}(n,8)+2$
3	$\text{mod}(n,8)+3$
4	$\text{mod}(n,8)+4$

3. Выберите в качестве выходной величины, определяющей состояние процесса переменную мониторинга с индексом 4.

4. Зададитесь числом состоянием по Y - 3, X_1 -5, X_2 - 3, X_3 -2.

5. Определите диапазон изменений состояний, причем для Y_1, X_1 , - автоматически по дисперсии (здесь и далее рекомендуется использовать интегрированные среды типа EXCEL, STATISTICA), X_2 -- экспертным путем анализа гистограммы, X_3 - экспериментальным путем заданием одного порога ($<, >=$).

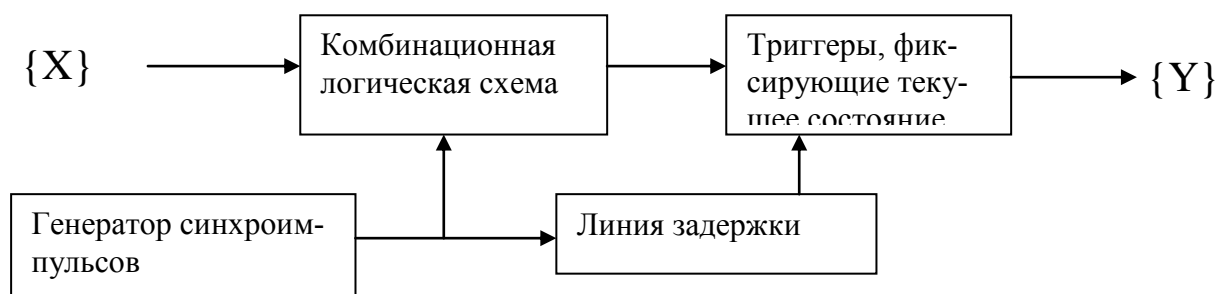
6. Закодируйте состояние по X, Y - т.е. получите характеристические числа (вид) булевых функций X_6 и Y_6 .

7. Проанализируйте взаимозависимость между булевыми факторами X и сформируйте наиболее ортогональное факторное пространство (векторы которого в наименьшей степени зависимы между собой).

8. Идентифицируйте в минимальном виде (с помощью карт Карно) логические функции $Y_6 = F(X_6)$.

9. Перейдите от булевского представления к логико-семантическому и сформулируйте диагностические правила продукционного типа.

10. Составьте схмотехническое решение диагностических правил по следующей структуре:



Рассчитайте скважность синхроимпульсов и их характер для обеспечения устойчивой работы схем, считая время срабатывания любого логического элемента 10мс, время срабатывания триггера 50мс.

11. Оформите отчет с указанием последовательности своих действий, необходимых комментариев и выводов.

Контрольные вопросы:

1. Что определяет решающее правило?
2. Какие типы решающих правил применяют в диагностическом процессе при обработке результатов мониторинга?
3. В чем заключается логический способ синтеза решающего правила?
4. Каким образом осуществляется бинарное кодирование признаков пространства при синтезе логических решающих правил?
5. Как формулируется решающее правило продукционного типа?
6. Как осуществляется семантическое описание решающего правила?
7. Каким образом реализуется схемотехническая реализация решающего правила на определенной электронной базе?
8. Как проверяется качество применения решающего правила?

Таблица 1. Результаты регионального мониторинга (начало)

год	Всего родилось	Всего заболело	Врожденные пороки (ВП)	асфаксия	Умерло всего	Умерло от ВП	Умерло от асфиксии
1	1657	90	22	6	17	4	5
2	2081	170	24	9	32	3	9
3	2173	201	20	25	32	7	5
4	2676	198	41	17	34	5	5
5	2557	191	51	47	21	3	10
6	2522	586	83	78	23	1	3
7	2893	252	30	19	31	3	7
8	2956	270	45	8	32	6	5
9	2650	197	38	12	25	3	6
10	3036	213	42	36	36	4	12
11	3165	230	37	32	21	2	5
12	3181	218	61	42	27	8	3
13	2930	216	65	58	18	1	6
14	2491	202	41	55	20	2	5
15	2964	185	37	39	25	2	1
16	2425	290	65	87	22	2	7
17	2432	238	50	53	19	3	4
18	2388	196	34	65	19	0	9
19	2290	197	34	58	22	4	9
20	2995	193	45	27	28	5	7

Таблица 1. Результаты регионального мониторинга (окончание)

Год	Ревм. пораж. сердца	Инфаркт миокарда	Гипертан. болезнь	стенокардия	Септический эндокартит	летальность
1	107	8	171	30	5	18
2	147	4	151	38	9	20
3	146	22	124	42	4	17
4	122	27	145	56	10	20
5	104	37	134	83	8	23
6	77	37	110	33	11	23
7	82	24	156	38	8	20
8	104	37	100	40	4	13
9	88	21	87	45	35	18
10	75	17	111	41	6	2
11	44	28	71	29	3	17
12	56	4	120	38	8	12
13	44	16	100	31	7	7
14	44	12	104	28	5	24
15	57	15	97	35	15	10
16	32	17	84	33	14	5
17	36	15	116	40	20	16
18	44	20	176	44	2	28
19	51	23	157	37	11	13
20	48	18	162	43	17	17

ЛАБОРАТОРНАЯ РАБОТА № 6

ИССЛЕДОВАНИЕ ИЗМЕНЕНИЯ СОСТОЯНИЯ ОБЪЕКТА В ПРОСТРАНСТВЕ СОСТОЯНИЙ

Цель работы: Освоить применение программных средств автоматизированной технологии отображения и исследования изменения местонахождения исследуемого объекта в пространстве состояний (факторном пространстве).

Краткие теоретические сведения.

Исследуемый процесс (объект) в определенный момент времени характеризуется значением вектора регистрируемых характеристик в многомерном гиперпространстве состояний, где он в это момент может рассматриваться точкой с координатами (x_1, x_2, \dots, x_m) . Заметим, что ортогональность m «осей» не является, в общем случае, обязательным условием.

Движение точки во времени представляет собой определенный профиль-диаграмму, по которому можно, с одной стороны - прогнозировать процесс, с другой - классифицировать его состояния, с третьей - идентифицировать функциональную зависимость «движения» во времени. Однако, если размерность пространства больше трех, то визуальный анализ затруднен, а значит и проблематично использование экспертного анализа и возможностей когнитивной графики. Поэтому прибегают к сужению размерности факторного пространства до двух, реже до трех псевдофакторов с целью отображения динамики процесса на плоскости.

К настоящему времени существует множество процедур конструирования двумерных изображений, однако практически все они не имеют обратных функций, т.е. не являются взаимнообратимыми.

Рассмотрим способ, сохраняющий подобие расстояний, т.е. по крайней мере сохраняющий при переходе характер близости отображающих точек.

Итак, пусть процесс представлен следующей таблицей наблюдений X_{nm} , где n - количество наблюдений, m - количество регистрируемых характеристик.

Выберем в исходном многомерном пространстве три опорных точки 1,2,3. Выбор может быть осуществлен случайным образом, но первоначально рекомендуется поступать следующим образом:

1) провести парный корреляционный анализ между характеристиками $x_1 \dots x_m$ и выбрать два наибольших ортогональных из них: S , R .

2) Рассчитать оценки математического ожидания (среднее) и дисперсии x_c , x_r (а также математические ожидания остальных характеристик).

3) в качестве опорных выбрать точки с координатами:

$$x_{01} = (M(x_1), (x_2) \dots M(x_e) \dots M(x_n), \dots M(x_m))$$

$$x_{02} = (M(x_1), (x_2) \dots M(x_e) + (x_e) \dots M(x_k) + (x_k))$$

$$x_{03} = (M(x_1), (x_2) \dots M(x_e) - (x_e) \dots M(x_k) - (x_k))$$

где $M(x_i)$ - среднее значение i -ой характеристики,

(x_i) - дисперсия i -той характеристики.

Заметим, что в случае нормирования исходной информации по дисперсии имеем:

$$x_{01} = (0, 0, \dots, 0, 0, \dots, 0);$$

$$x_{02} = (0, 0, \dots, 1, 1, \dots, 0);$$

$$x_{03} = (0, 0, \dots, -1, -1, \dots, 0);$$

Для отображения точки в пространстве R^2 воспользуемся теоремой Пифагора и правилом, что три окружности пересекаются только в одной точке. Итак, пусть необходимо отобразить некоторую точку z в R^n в пространстве R^2 .

На первом этапе определяем расстояния точки z до опорных в R^n :

На втором этапе строим в R^2 систему координат следующим образом:

- 1) точку X_{01} помещаем в начало координат;
- 2) точка X_{02} образует с предыдущей точкой ось ординат с сохранением расстояния;
- 3) точка X_{03} также сохраняет расстояния между второй и третьей опорными точками.

Определим (применяя теорему Пифагора) координаты точки X_z в R^2 с учетом сохранения расстояний:

Т.о. происходит последовательное отображение каждой точки z .

В случае если профиль недостаточно нагляден - изменяются базовые координаты опорных точек 1 и 2. Кроме того, в R^2 можно применять линейные преобразования пространства для улучшения изображения.

Порядок выполнения работы.

1. Самостоятельно изучить теоретический материал.
2. Инициализируйте программное обеспечение автоматизированной системы RMR2.EXE. Введите информацию Таблицы 1, выделив 5-6 состояний (например, как в лабораторной работе № 2).
3. Варьируя опорные координаты опорных точек и линейными преобразованиями, добейтесь наилучшего изображения, минимизируя «за-

липание» точек и максимизируя расстояния между образами различных состояний. Зафиксируйте это положение.

4. Оформите отчет, в который должно входить:

- краткие теоретические сведения;
- результат отображения последовательного соединения образов;
- зафиксированный вариант отображения (координаты и характер линейного преобразования);
- последовательность ваших действий по работе с программой RMR2.EXE.
- ВЫВОДЫ.

Таблица 1

n	x1	x2	x3	x4	x5	x6	x7	x8
месяц								
1	31.67	91.23	138	39	2.31	2.9	17.2	6.7
2	30.88	94.08	111	34	2.36	2.3	16.5	6
3	28.66	88.31	108	33	2.33	3.0	19.8	4.9
4	28.79	88.21	113	35	2.10	3.2	19.7	5.1
5	28.67	88.42	113	35	1.93	2.2	18.0	5
6	28.14	90.51	114	36	1.80	2.9	17.2	4.1
7	26.53	85.59	108	35	1.74	2.7	21.2	4.5
8	26.13	87.45	104	35	1.92	2.2	21.3	5.1
9	26.12	84.73	110	36	1.71	3.2	21.7	4.5
10	26.81	88.02	108	35	1.72	2.1	20.3	4.2
11	25.96	85.54	106	35	1.80	2.1	21.6	3.8
12	26.36	82.49	114	36	1.64	3.3	22.6	3.9
годы								
2	26.90	80.43	118	35	1.75	2.7	31.7	4.5
3	28.23	91.25	119	38	1.27	3.0	34.5	3.4
4	29.35	92.78	121	38	0.97	5.0	34.4	3.6
5	29.63	94.45	122	38	0.87	4.1	33.8	4.1
6	29.05	84.90	127	37	1.09	4.1	37.6	4.6
7	29.36	89.92	128	36	1.12	2.9	41.1	4.6

Где x_1 - среднее содержание гемоглабина в эритроците,
 x_2 - средний объем эритроцитов,
 x_3 – гемоглабин,
 x_4 – гемотакрит,
 x_5 – ретикулоциты,
 x_6 - нейтрофилы палочкоядерные,
 x_7 - нейтрофилы сегментоядерные,
 x_8 – моноциты.

Вопросы:

1. Зачем прибегают к сужению мерности факторного пространства?
2. Как может задаваться в многомерном гиперпространстве состояние объекта?
3. Какие способы визуализации состояния объекта вы знаете?

ЛАБОРАТОРНАЯ РАБОТА №7

Графический интерфейс гибридных систем в Matlab

Цель работы. Целью данной практической работы является знакомство с гибридными системами, реализованными в прикладном программном пакете Matlab и приобретение навыков по построению гибридных систем.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Графический интерфейс гибридных (нечетких) нейронных систем вызывается функцией (из режима командной строки) **anfisedit**. Исполнение функции приводит к появлению окна редактора гибридных систем (ANFIS Editor, ANFIS-редактор), вид которого приведен на рис. 1.

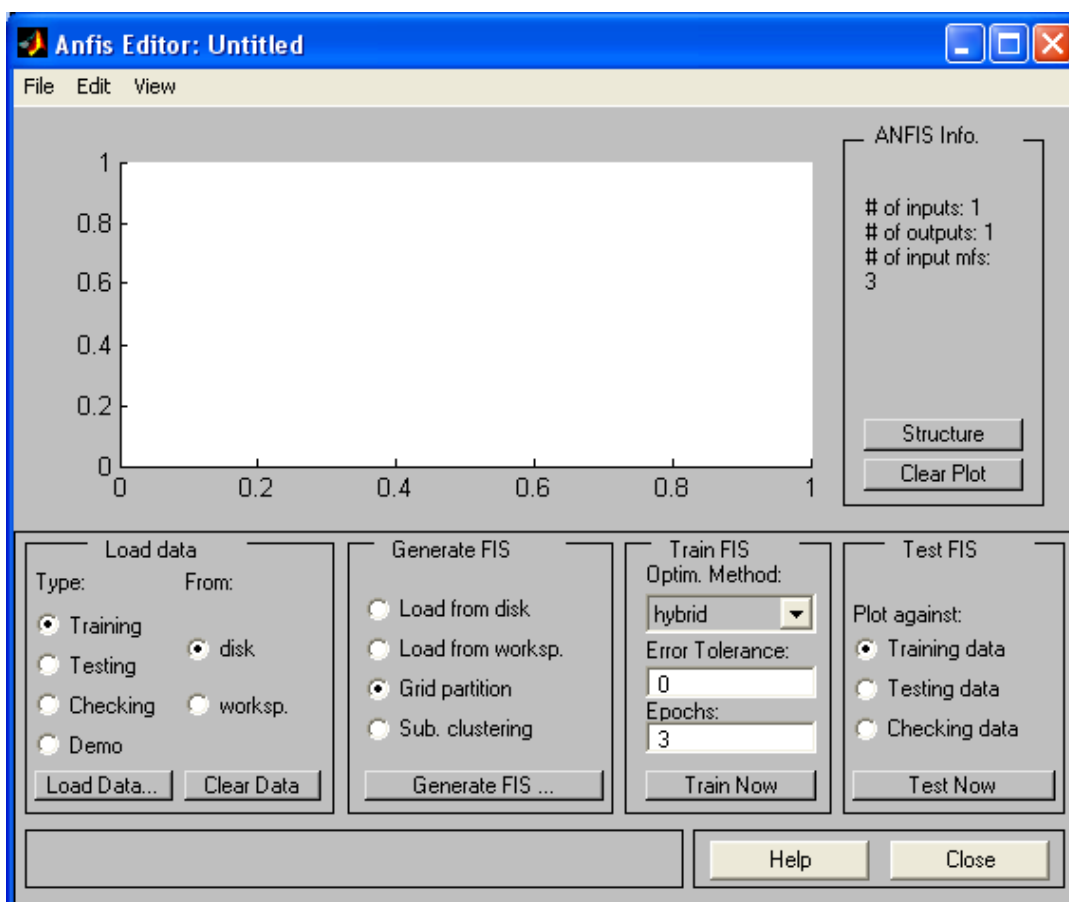


Рис. 1. Окно редактора гибридных систем

С помощью данного редактора осуществляется создание или загрузка структуры гибридной системы, просмотр структуры, настройка ее параметров, проверка качества функционирования такой системы. Со-

здание структуры и настройка параметров и проверка осуществляются по выборкам (наборам данных) — обучающей (Training data), проверочной (Checking data) и тестирующей (Testing data), которые предварительно должны быть представлены в виде текстовых файлов (с расширением .dat и разделителями-табуляциями), первые колонки которых соответствуют входным переменным, а последняя (левая) — единственной выходной переменной; количество строк в таких файлах равно количеству образцов (примеров).

Так, обучающая выборка, сформированная по табл. 1, представляется в виде

-1	1
-0.6	0.36
0.0	0.00
0.4	0.16
1	1

Строгих рекомендаций по объемам указанных выборок не существует, по-видимому, лучше всего исходить из принципа «чем больше, тем лучше». Обучающая и проверочная выборки непосредственно задействуются в процессе настройки параметров гибридной сети (проверочная — для выяснения ситуации: нет ли так называемого переобучения сети, при котором ошибка для обучающей последовательности стремится к нулю, а для проверочной — возрастает; впрочем, наличие проверочной выборки не является строго необходимым, оно лишь крайне желательно). Тестовая (или тестирующая выборка) применяется для проверки качества функционирования настроенной (обученной) сети.

Поясним пункты меню и опции редактора.

Пункты меню File и View, в общем идентичны аналогичным пунктам FIS-редактора, за тем исключением, что здесь работа может происходить только с алгоритмом нечеткого вывода Sugeno. Пункт меню Edit содержит единственный подпункт — Undo (Отменить выполненное действие).

Набор опций Load data (Загрузить данные) в нижней левой части окна редактора включает в себя:

- тип (Type) загружаемых данных (для обучения — Training, для тестирования — Testing, для проверки — Checking, демонстрационные — Demo);
- место, откуда должны загружаться данные (с диска — disk или из рабочей области MATLAB-workspace).

К данным опциям относятся две кнопки, нажатие на которых приводит к требуемым действиям — Load Data... (Загрузить данные) и Clear Data (очистить, т.е. стереть введенные данные).

Следующая группа опций (в середине нижней части окна ANFIS-редактора) объединена под именем **Generate FIS** (Создание нечеткой системы вывода). Данная группа включает в себя опции:

- загрузку структуры системы с диска (Load from disk);
- загрузку структуры системы из рабочей области MATLAB (Load from worksp.);
- разбиение (деление) областей определения входных переменных (аргументов) на подобласти — независимо для каждого аргумента (Grid partition);
- разбиение всей области определения аргументов (входных переменных) на подобласти — в комплексе для всех аргументов (Subtract clustering или Sub. clustering), а также кнопку Generate FIS, нажатие которой приводит к процессу создания гибридной системы с точностью до ряда параметров.

Следующая группа опций — **Train FIS** (Обучение нечеткой системы вывода) — позволяет определить метод «обучения» (Optim. Method) системы (т.е. метод настройки ее параметров) — гибридный (hybrid) или обратного распространения ошибки (back-програ), установить уровень текущей суммарной (по всем образцам) ошибки обучения (Error Tolerance), при достижении которого процесс обучения заканчивается и количество циклов обучения (**Epochs**), т.е. количество «прогонов» всех образцов (или примеров) обучающей выборки; процесс обучения, таким образом заканчивается либо при достижении отмеченного уровня ошибки обучения, либо при проведении заданного количество циклов.

Кнопка **Train Now** (Начать обучение) процесс обучения, т.е. процесс настройки параметров гибридной сети.

В правом верхнем углу окна ANFIS-редактора выдается информация (ANFIS Info.) о проектируемой системе: о количестве входов, выходов, функций принадлежности входов; нажатие кнопки Structure (Структура) позволяет увидеть структуру сети. Кнопка Clear (Очистить) позволяет стереть все результаты.

Опции **Test FIS** в правом нижнем углу окна позволяют провести проверку и тестирование созданной и обученной системы с выводом результатов в виде графиков (соответствующие графики для обучающей выборки — Training data, тестирующей выборки — Testing data и проверочной

выборки — Checking data. Кнопка Test Now позволяет запустить указанные процессы.

Работу с редактором рассмотрим на примере восстановления зависимости $y = x^2$ по данным табл. 1. Предположим, что эти данные сохранены в файле Proba.dat. Создание и проверку системы, как и раньше, проведем по этапам.

1. В окне ANFIS-редактора выберем тип загружаемых данных Training и нажмем кнопку Load data. В последующем стандартном окне диалога укажем местоположение и имя файла. Его открытие приводит к появлению в графической части окна редактора набора точек, соответствующих введенным данным (рис. 2).

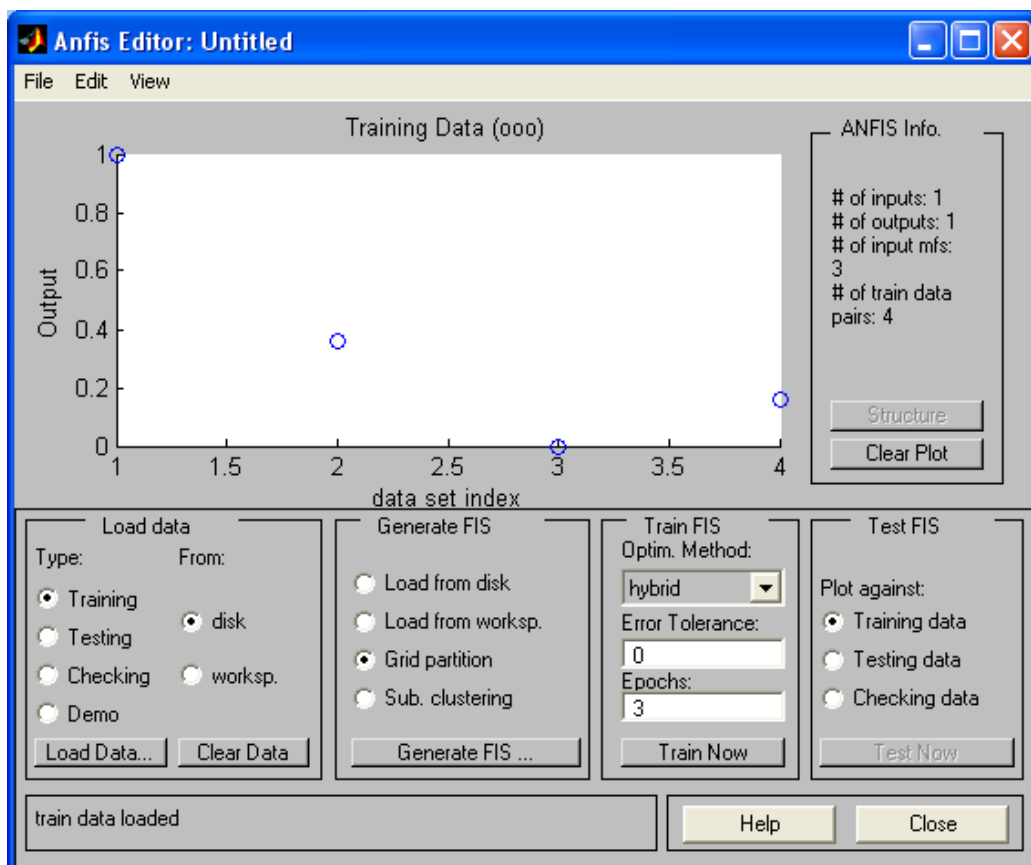


Рис.2 – Окно редактора после загрузки обучающей выборки

2. В группе опций Generate FIS по умолчанию активизирована опция Grid partition. Не будем ее изменять и нажмем кнопку Generate FIS, после чего появится диалоговое окно (рис. 3) для задания числа и типов функций принадлежности.

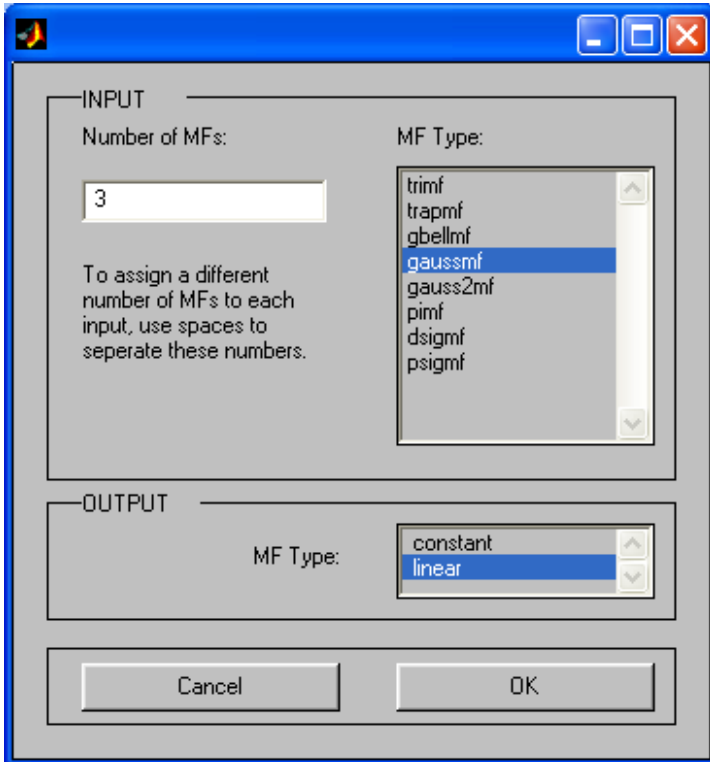


Рис.3 – Окно задания функций принадлежности

Сохраним все установки по умолчанию, согласившись с ними нажатием кнопки ОК. Произойдет возврат в основное окно ANFIS-редактора. Теперь структура гибридной сети создана, и ее графический вид можно просмотреть с помощью кнопки Structure (рис. 4).

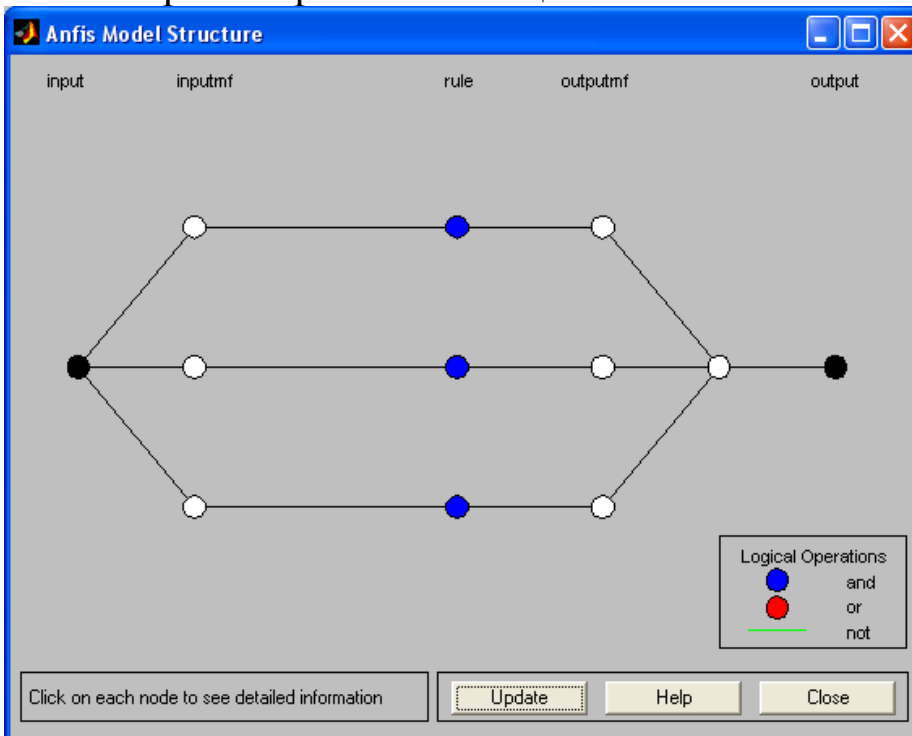


Рис.4 – Структура созданной гибридной сети

3.Перейдем к опциям Train FIS. Не будем менять задаваемые по умолчанию метод настройки параметров (hybrid — гибридный)и уровень ошибки (0), но количество циклов обучения изменим на 40, после чего нажмем кнопку начала процесса обучения (TrainNow). Получившийся результат в виде графика ошибки сети в зависимости от числа проведенных циклов обучения (из которого следует, что фактически обучение закончилось после пятого цикла)представлен на рис. 5.19.

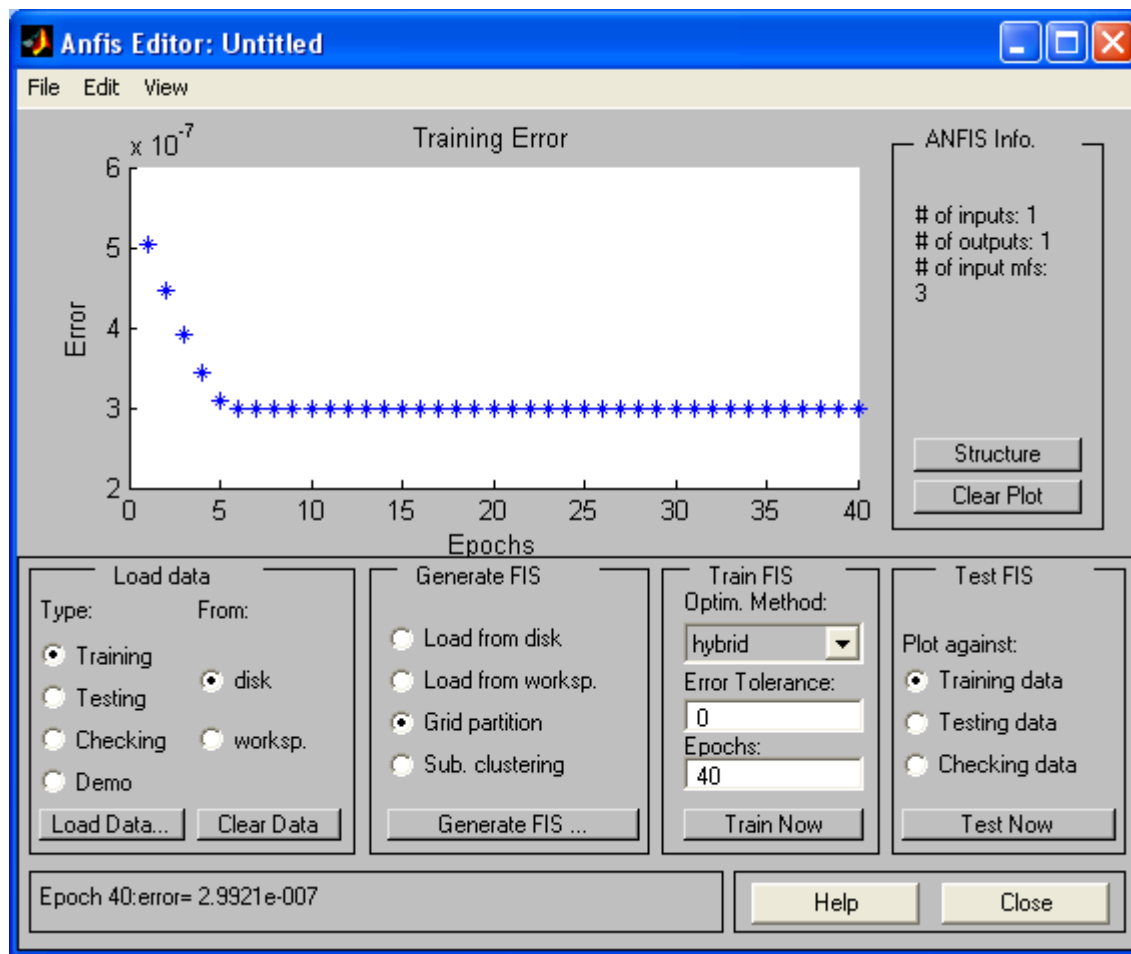


Рис.5 – Функция ошибки обучения сети

4.Теперь нажатием кнопки Test Now можно начать процесс тестирования обученной сети, но, поскольку использовалась только одна — обучающая — выборка, ничего особенно интересного ожидать не приходится. Действительно, выход обученной системы практически совпадает с точками обучающей выборки(рис.6).

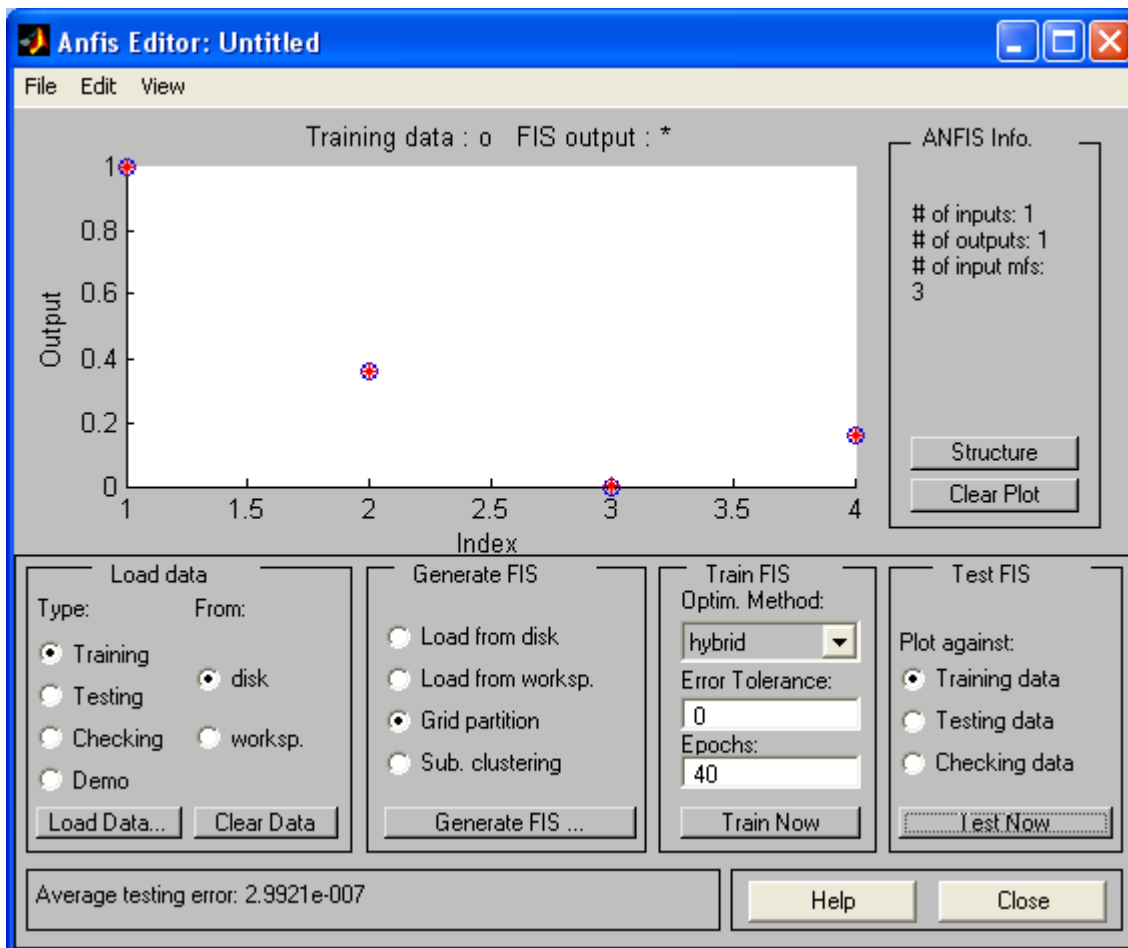


Рис. 6. Результат тестирования обученной системы

5. Сохраним разработанную систему в файл на диске с именем Probal (с расширением .Us) и для исследования разработанной системы средствами FIS-редактора из командной строки MATLAB выполним команду **fuzzy**, а затем через пункты меню File/Open FIS from disk... откроем созданный файл. С созданной системой можно теперь выполнять все приемы редактирования (изменение имен переменных и т. п.) и исследования, которые были рассмотрены выше. Здесь нетрудно, кстати, убедиться, что качество аппроксимации данных существенно не улучшилось — слишком мало данных.

Что можно сказать про эффективность использования гибридных систем (и ANFIS-редактора)?

В данном случае используется только один алгоритм нечеткого вывода — Sugeno (нулевого или первого порядков), может быть только одна выходная переменная, всем правилам приписывается один и тот же единичный вес. Вообще говоря, возникают значительные проблемы при большом (более 5-6) количестве входных переменных. Это — ограничения и недостатки подхода.

Его несомненные достоинства: практически полная автоматизация процесса создания нечеткой (гибридной) системы, возможность просмотр-

ра сформированных правил и придания им содержательной (лингвистической) интерпретации, что позволяет, кстати говоря, рассматривать аппарат гибридных сетей как средство извлечения знаний из баз данных и существенно отличает данные сети от классических нейронных.

Рекомендуемая область применения: построение аппроксиматоров зависимостей по экспериментальным данным, построение

систем классификации (в случае бинарной или дискретной выходной переменной), изучение механизма явлений.

Графический интерфейс программы кластеризации

В пакет Fuzzy Logic Toolbox входит еще одна программа, позволяющая работу в режиме графического интерфейса, — программа Clustering (Кластеризация) выявления центров кластеров, т.е. точек в многомерном пространстве данных, около которых группируются (скапливаются) экспериментальные данные. Выявление подобных центров, надо сказать, является значимым этапом при предварительной обработке данных, поскольку позволяет сопоставить с этими центрами функции принадлежности переменных при последующем проектировании системы нечеткого вывода.

Запуск программы Clustering осуществляется командой (функцией) **findcluster**. В появляющемся окне программы имеется (вверху) главное меню, содержащее достаточно стандартный набор пунктов (File, Edit, Window, Help) и набор управляющих кнопок и опций (справа). К этим кнопкам относятся:

- кнопка загрузки файла данных Load Data,
- кнопка выбора алгоритма кластеризации — Method,
 - четыре расположенные ниже кнопки опций алгоритма (их названия меняются в зависимости от выбранного алгоритма),
 - кнопка начала итеративного процесса нахождения центров кластеров (кластеризации) — Start,
- кнопка сохранения результатов кластеризации (SaveCenter),
- кнопка очистки (стирания) графиков (Clear Plot),
- кнопка справочной информации (Info),
- кнопка завершения работы с программой (Close).

В программе используются два алгоритма выявления центров кластеров: Fuzzy c-means (который можно перевести как «Алгоритм нечетких центров») и Subtractive clustering («Вычитающая кластеризация»).

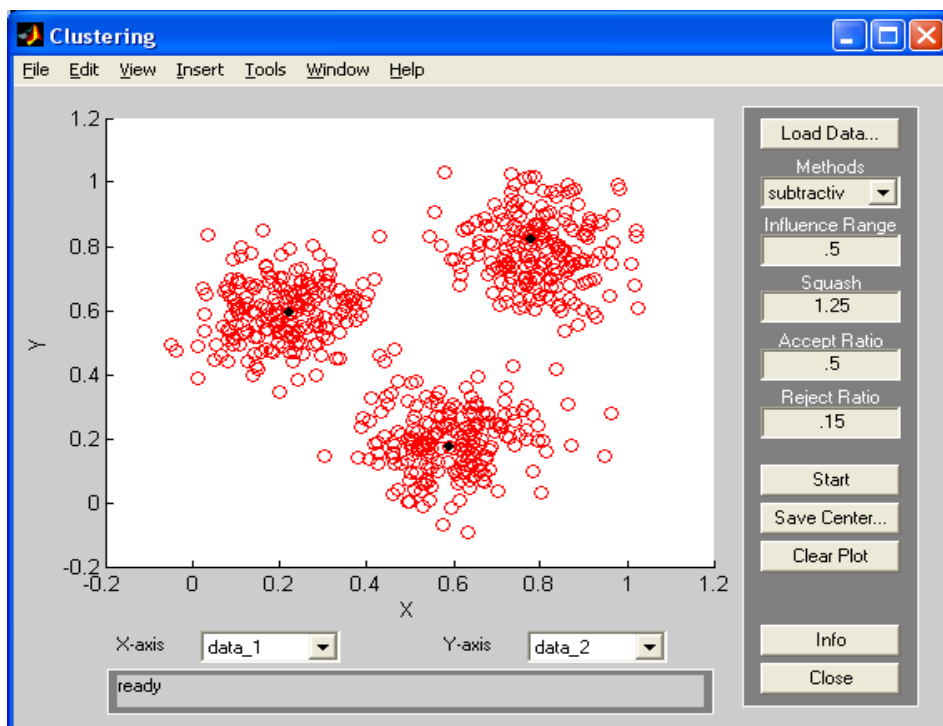


Рис. 7. Результат работы программы Clustering (центры кластеров окрашены в черный цвет)

Если не вдаваться в их детальное теоретическое изложение, а ограничиться выявлением различий на уровне пользователя, то можно отметить, что алгоритм Fuzzy c-means, являясь, пожалуй, более точным (если понятие точности вообще здесь применимо), для своей работы требует задания таких опций, как число кластеров (кнопка Cluster num.) и число итераций (кнопка Max Iteration[^]). Ну, если число итераций еще можно задать как-то наугад, то ошибка в задании числа кластеров может привести к неприятным последствиям. Алгоритм Subtractive clustering менее точен, но и менее требователен к априорной информации; при работе с ним можно сохранить опции, заданные в программе по умолчанию. На рис. 7 приведен пример использования программы для фала данных clusterdemo.dat из директории Matlab/toolbox/fuzzy/fuzdemos/ при использовании алгоритма Subtractive clustering. Заметим, что выводится только двумерное поле рассеяния, но изменяя переменные в соответствующих полях (X-axis и Y-axis), можно «просмотреть» все многомерное пространство переменных.

Индивидуальные задания

Аппроксимировать при помощи гибридной сети уравнения вида $y=f(x)$ по десяти точкам.

Таблица

№	f(x)
1	$\ln(x) - 1/(1 + x^2)$
2	$\ln(\ln(x)) - x^2$
3	$x - 1 / \sqrt{e^x}$
4	$x^4 - 13x^2 + 36 - (1 / x)$
5	$2x^2 - x^4 - 1 - \ln(x)$
6	$x^3 - 3x - 2e^{-x}$
7	$\sin(x^2) - 6x + 1$
8	$\cos(x^2) - 10x$
9	$\ln^2 x - (1 / x)$
10	$\ln(1 + x) / (1 - x) - \cos(x^2)$
11	$x - 1 / (x^4 - 13x^2 + 36)$
12	$e^x - 3 - \cos(x)$

Отчёт по лабораторной работе включает в себя:

- титульный лист;
- листы выполнения работы (цель работы, практическая часть, вывод, список литературы).

Практическая часть сопровождается таблицами, графиками с кратким пояснением хода выполнения лабораторной работы. Обязательно приводится описание параметров модели, можно распечатать экранные формы с пояснениями.

Контрольные вопросы

1. Какие функции выполняет графический интерфейс гибридных (нечетких) нейронных систем?
2. Поясните основной функционал окна задания функций принадлежности.
3. Что показывает график ошибки обучения сети?
4. Какие функции выполняет графический интерфейс программы кластеризации?

Литература

1. Осовский С., Нейронные сети для обработки информации / Пер. с польского И.Д.Рудинского. – М.: Финансы и статистика, 2002.
2. Джонс М.Т. Программирование искусственного интеллекта в приложениях. - М. ДМК Пресс, 2004.

ЛАБОРАТОРНАЯ РАБОТА № 8
ПОСТРОЕНИЕ НЕЧЕТКОЙ СИСТЕМЫ И ИСПОЛЬЗОВАНИЕМ ПАКЕТА
Fuzzy Logic Toolbox

Цель работы: знакомство с использованием пакета Fuzzy Logic Toolbox, реализованным в прикладном программном пакете Matlab и приобретение навыков по построению нечетких систем управления.

Краткие теоретические сведения

В последнее время нечеткая технология завоевывает все больше сторонников среди разработчиков систем управления. Термин «fuzzy» так вошел в жизнь, что на многих языках он даже не переводится. В России в качестве примера можно вспомнить рекламу стиральных машин и микроволновых печей фирмы Samsung, обладающих искусственным интеллектом на основе нечеткой логики. Тем не менее, столь масштабный скачок в развитии нечетких систем управления не случаен. Простота и дешевизна их разработки заставляет проектировщиков все чаще прибегать к этой технологии.

Нечеткая логика основана на использовании таких оборотов естественного языка, как «далеко», «близко», «холодно», «горячо». Диапазон ее применения очень широк - от бытовых приборов до управления сложными промышленными процессами. Многие современные задачи управления просто не могут быть решены классическими методами из-за очень большой сложности математических моделей, их описывающих. Вместе с тем, чтобы использовать теорию нечеткости на цифровых компьютерах, необходимы математические преобразования, позволяющие перейти от лингвистических переменных к их числовым аналогам в ЭВМ.



Рисунок 1 Области применения технологий управления

На рисунке 1 показаны области наиболее эффективного применения современных технологий управления. Как видно, классические методы управления хорошо работают при полностью детерминированном объекте управления и детерминированной среде, а для систем с неполной информацией и высокой сложностью объекта управления оптимальными являются нечеткие методы управления.

Логика известна нам как строгая и сугубо теоретическая наука, и большинство ученых (кроме разработчиков последнего поколения компьютеров) продолжают придерживаться этого мнения. Вместе с тем классическая или булева логика имеет один существенный недостаток - с ее помощью невозможно описать ассоциативное мышление человека. Классическая логика оперирует только двумя понятиями: ИСТИНА и ЛОЖЬ, исключая любые промежуточные значения. Аналогично этому булева логика не признает ничего кроме единиц и нулей. Все это хорошо для вычислительных машин, но недостаточно эффективно при проектировании и эксплуатации систем управления БТС.

Решить эту проблему и призвана нечеткая логика. С термином «лингвистическая переменная» можно связать любую физическую величину, для которой нужно иметь больше значений, чем только ДА и НЕТ. В этом случае вы определяете необходимое число термов и каждому из них ставите в соответствие некоторое значение описываемой физической величины. Для этого значения степень принадлежности физической величины

к терму будет равна единице, а для всех остальных значений - в зависимости от выбранной функции принадлежности.

Получившие наибольшее развитие из всех разработок искусственного интеллекта, экспертные системы завоевали устойчивое признание в качестве систем поддержки принятия решений. Одним из основных методов представления знаний в экспертных системах являются продукционные правила, позволяющие приблизиться к стилю мышления человека. Любое правило продукций состоит из посылок и заключения. Главным же недостатком продукционных систем остается то, что для их функционирования требуется наличие полной информации о системе.

Нечеткие системы тоже основаны на правилах продукционного типа, однако в качестве посылки и заключения в правиле используются лингвистические переменные, что позволяет избежать ограничений, присущих классическим продукционным правилам.

Целевая установка процесса управления связывается с выходной переменной нечеткой системы управления, но результат нечеткого логического вывода является нечетким, а физическое исполнительное устройство не способно воспринять такую команду. Необходимы специальные математические методы, позволяющие переходить от нечетких значений величин к вполне определенным. В целом весь процесс нечеткого управления можно разбить на несколько шагов: фаззификация, разработка нечетких правил и дефаззификация.

Фаззификация (переход к нечеткости)

Точные значения входных переменных преобразуются в значения лингвистических переменных посредством применения некоторых положений теории нечетких множеств, а именно - при помощи определенных функций принадлежности.

Лингвистические переменные

В нечеткой логике значения любой величины представляются не числами, а словами естественного языка и называются ТЕРМАМИ. Так, значением лингвистической переменной ДИСТАНЦИЯ являются термы ДАЛЕКО, БЛИЗКО и т. д.

Функции принадлежности

Как уже говорилось, принадлежность каждого точного значения к одному из термов лингвистической переменной определяется посредством

функции принадлежности. Ее вид может быть абсолютно произвольным. Стандартные функции принадлежности представлены на рисунке 2.



Рисунок 2. – типовые функции принадлежности

Стандартные функции принадлежности легко применимы к решению большинства задач.

Разработка нечетких правил

На этом этапе определяются продукционные правила, связывающие лингвистические переменные. Совокупность таких правил описывает стратегию управления, применяемую в данной задаче. Большинство нечетких систем используют продукционные правила для описания зависимостей между лингвистическими переменными. Типичное продукционное правило состоит из антецедента (часть ЕСЛИ ...) и консеквента (часть ТО ...). Антецедент может содержать более одной посылки. В этом случае они объединяются посредством логических связок И или ИЛИ. Процесс вычисления нечеткого правила называется нечетким логическим выводом и подразделяется на два этапа: обобщение и заключение.

Дефаззификация (устранение нечеткости)

На этом этапе осуществляется переход от нечетких значений величин к определенным физическим параметрам, которые могут служить командами исполнительному устройству. Результат нечеткого вывода, конечно же, будет нечетким.

Для устранения нечеткости окончательного результата существует несколько методов. Рассмотрим некоторые из них.

Метод центра максимума (CoM)

Так как результатом нечеткого логического вывода может быть несколько термов выходной переменной, то правило дефаззификации должно определить, какой из термов выбрать.

Метод наибольшего значения (MoM)

При использовании этого метода правило дефаззификации выбирает максимальное из полученных значений выходной переменной.

Метод центроида (CoA)

В этом случае окончательное значение определяется как проекция центра тяжести фигуры, ограниченной функциями принадлежности выходной переменной с допустимыми значениями.

Основные шаги разработки нечеткой системы управления с использованием САД-системы fuzzy TECH 3.0

Процесс разработки проекта нечеткой системы управления на fuzzy TECH разбивается, как уже говорилось, на четыре основных этапа. Все они схематично показаны на рисунке 3.



Рисунок 3 – основные этапы разработки нечеткой системы управления.

Описание системы

На этом этапе при помощи средств, доступных в fuzzy TECH, задача формализуется. Здесь описываются: лингвистические переменные, их функции принадлежности, описать стратегия управления посредством нечетких правил. В целом CASE-технология, на основе которой построен

пакет, позволяет все эти действия выполнить только посредством общения с экраном ЭВМ, не заглядывая в программный код.

Off-line-оптимизация

На этом этапе следует проверить работоспособность созданной системы посредством всех средств fuzzy ТЕСН. Отметим, что можно использовать заранее созданный программный симулятор вашего объекта управления, подобно модели контейнерного крана. Для связи системы управления с моделью используется специально разработанный протокол связи fTlink, в основу которого положена концепция обмена сообщениями Windows. Все необходимые средства для установления связи с вашей моделью находятся в исходных текстах программ связи, поставляемых с пакетом.

On-line-оптимизация

На этом шаге разрабатываемая система управления и реальный объект управления соединяются физической линией связи (см. рисунок 4)



Рисунок 4 – система управления при On-line-оптимизации

Такой вид отладки позволяет наблюдать поведение системы в реальных условиях и при необходимости вносить изменения в систему управления.

ФУНКЦИИ ПО РАБОТЕ С ПАКЕТОМ FUZZY LOGIC TOOLBOX

Редактор функций принадлежности

Редактор функций принадлежности-(**Membership Function Editor**) редактор предназначен для задания следующей информации о термножествах входных и выходных переменных: количество термов; наименования термов; тип и параметры функций принадлежности, кото-

рые необходимы для представления лингвистических термов в виде нечетких множеств.

Редактор функций принадлежности может быть вызван из любого GUI-модуля (графического модуля среды Matlab), используемого с системами нечеткого логического вывода, командой **Membership Functions...** меню **Edit** или нажатием клавиш **Ctrl+2**.

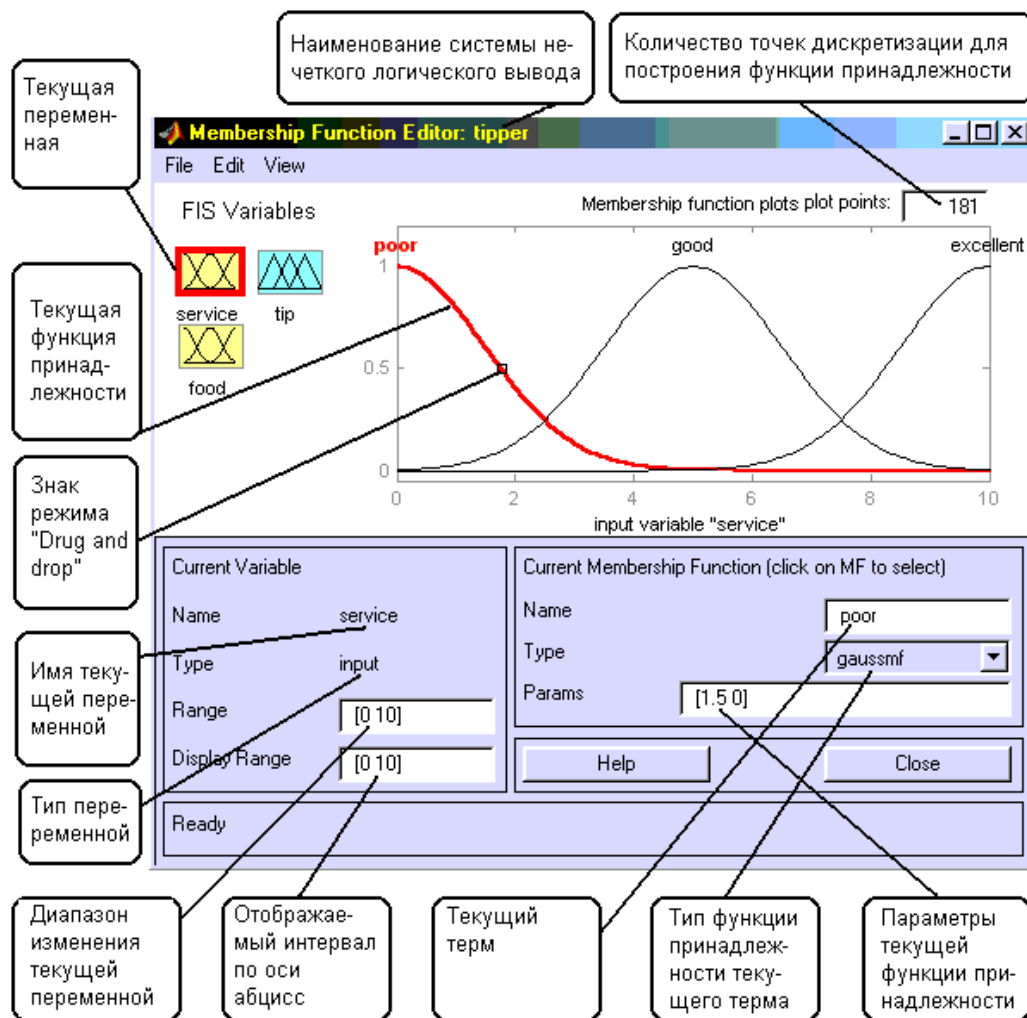


Рисунок 5 – основное диалоговое окно

В FIS-редакторе открыть редактор функций принадлежности можно также двойным щелчком левой кнопкой мыши по полю входной или выходной переменных. Общий вид редактора функций принадлежности с указанием функционального назначения основных полей графического окна приведен на рис. В нижней части графического окна расположены кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Редактор функций принадлежности содержит четыре меню - **File**, **Edit**, **View**, **Type** и четыре окна ввода информации – **Range**, **Display Range**,

Name и **Params**. Эти четыре окна предназначены для задания диапазона изменения текущей переменной, диапазона вывода функций принадлежности, наименования текущего лингвистического термина и параметров его функции принадлежности, соответственно. Параметры функции принадлежности можно подбирать и в графическом режиме, путем изменения формы функции принадлежности с помощью технологии “Drug and drop”. Для этого необходимо позиционировать курсор мыши на значке режима “Drug and drop” (см. рис.), нажать на левую кнопку мыши и не отпуская ее изменять форму функции принадлежности. Параметры функции принадлежности будут пересчитываться автоматически.

Меню Edit.

Общий вид меню.



Рисунок 6 - меню **Edit**

Команда **Undo** отменяет ранее совершенное действие. Выполняется также по нажатию **Ctrl+Z**. Команда **Add MFs...** позволяет добавить термины в терм-множество, используемое для лингвистической оценки текущей переменной. При выборе этой команды появится диалоговое окно, в котором необходимо выбрать тип функции принадлежности и количество терминов. Значения параметров функций принадлежности будут установлены автоматически таким образом, чтобы равномерно покрыть область определения переменной, заданной в окне **Range**. При изменении области определения в окне **Range** параметры функций принадлежности будут промасштабированы.

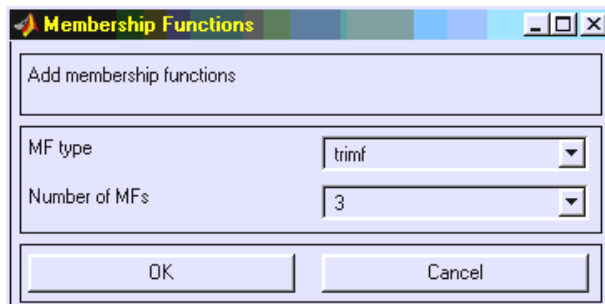


Рисунок 7 - выбор количества терминов и типа функций принадлежности.

Команда **Add Custom MF...** позволяет добавить один лингвистический терм, функция принадлежности которого отличается от встроенных. После выбора этой команды появится графическое окно, в котором необходимо напечатать лингвистический терм (поле **MF name**), имя функции принадлежности (поле **M-File function name**) и параметры функции принадлежности (поле **Parameter list**).

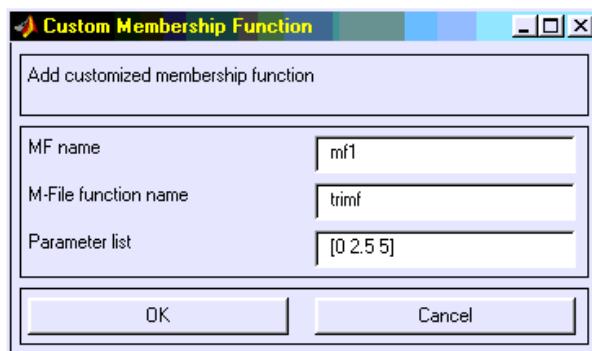


Рисунок 8 - задание лингвистического термина с невстроенной функцией принадлежности.

Команда **Remove Selected MF** удаляет текущий терм из терм-множества текущей переменной. Признаком текущей переменной является красная окантовка ее прямоугольника. Признаком текущего термина является красный цвет его функции принадлежности. Для выбора текущего термина необходимо провести позиционирования курсора мыши на графике функции принадлежности и сделать щелчок левой кнопкой мыши.

Команда **Remove All MFs** удаляет все термы из терм-множества текущей переменной. Команда **FIS Properties...** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Rules...** открывает редактор базы знаний. Эта команда может быть также выполнена нажатием **Ctrl+3**.

Меню Type

Это меню позволяет установить тип функций принадлежности термов, используемых для лингвистической оценки текущей переменной. На рис. приведено меню **Type**, в котором указаны возможные типы функций принадлежности.

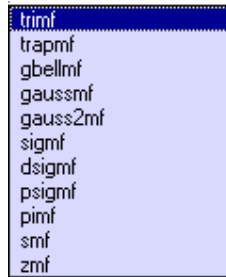


Рисунок 9 - Меню Type

Редактор базы знаний

Редактор базы знаний (**Rule Editor**) предназначен для формирования и модификации нечетких правил. Редактор базы знаний может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **Rules...** меню **Edit** или нажатием клавиш **Ctrl+3**. В FIS-редакторе открыть редактор базы знаний можно также двойным щелчком левой кнопкой мыши по прямоугольнику с названием системы нечеткого логического вывода, расположенного в центре графического окна.

Общий вид редактора базы знаний с указанием функционального назначения основных полей графического окна приведен на рис. В нижней части графического окна расположены кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Редактор функций принадлежности содержит четыре системных меню **File**, **Edit**, **View**, **Options**, меню выбора термов входных и выходных переменных, поля установки логических операций И, ИЛИ, НЕ и весов правил, а также кнопки редактирования и просмотра правил.

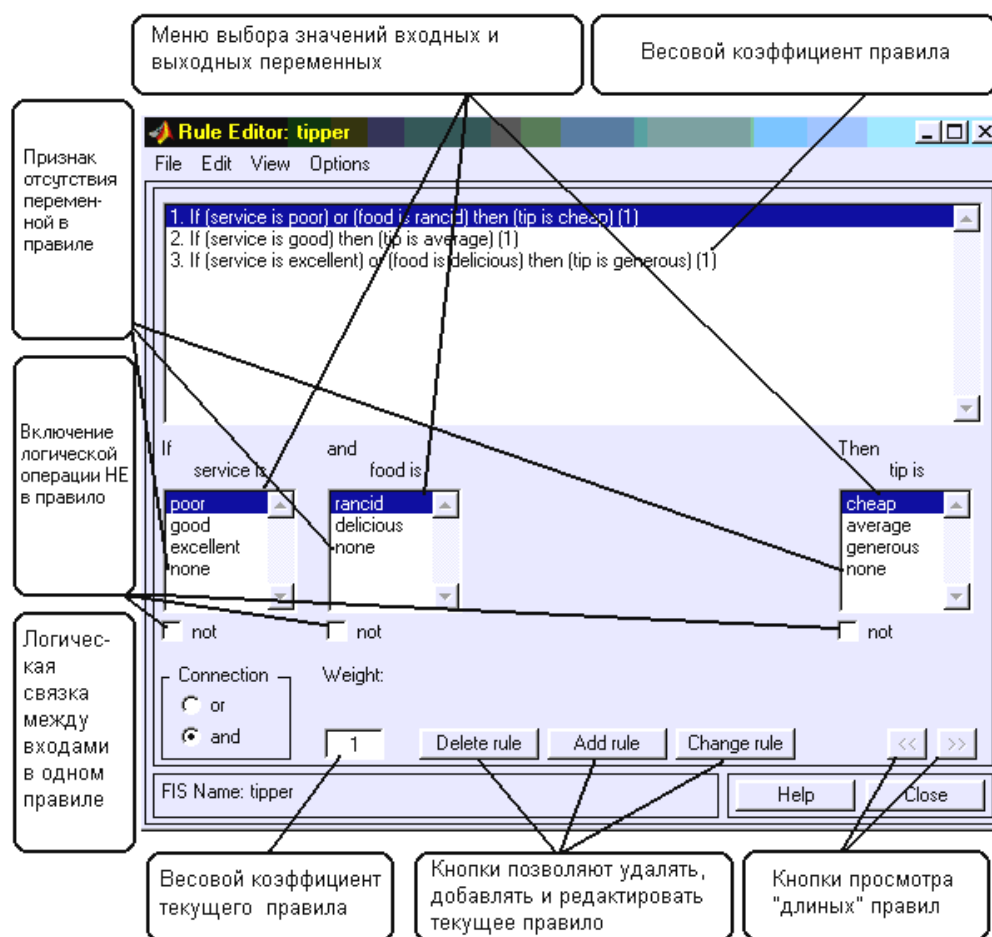


Рисунок 10 – основное диалоговое окно редактора база знаний

Для ввода нового правила в базу знаний необходимо с помощью мыши выбрать соответствующую комбинацию лингвистических термов входных и выходных переменных, установить тип логической связки (И или ИЛИ) между переменными внутри правила, установить наличие или отсутствие логической операции НЕ для каждой лингвистической переменной, ввести значение весового коэффициента правила и нажать кнопку **Add Rule**. По умолчанию установлены следующие параметры: логическая связка переменных внутри правила – И; логическая операция НЕ – отсутствует; значение весового коэффициента правила – 1.

Возможны случаи, когда истинность правила не изменяется при произвольной значении некоторой входной переменной, т.е. это переменная не влияет на результат нечеткого логического вывода в данной области факторного пространства. Тогда в качестве лингвистического значения этой переменной необходимо установить **none**.

Для удаления правила из базы знаний необходимо сделать однократный щелчок левой кнопкой мыши по этому правилу и нажать кнопку **Delete Rule**.

Для модификации правила необходимо сделать однократный щелчок левой кнопкой мыши по этому правилу, затем установить необходимые параметры правила и нажать кнопку **Edit Rule**.

Меню Edit

Общий вид меню приведен на рисунке 11.

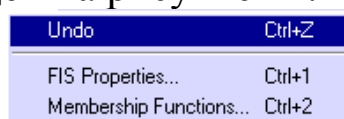


Рисунок 11 - меню **Edit**

Команда **Undo** отменяет ранее совершенное действие. Выполняется также по нажатию **Ctrl+Z**. Команда **FIS Properties...** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Membership Function...** открывает редактор функций принадлежности. Эта команда может быть также выполнена нажатием **Ctrl+2**.

Меню Options

Это меню позволяет установить язык и формат правил базы знаний.

При выборе команды **Language** появится список языков **English** (Английский), **Deutsch** (Немецкий), **Francais** (Французский), из которого необходимо выбрать один. При выборе команды **Format** появится список возможных форматов правил базы знаний: **Verbose** - лингвистический; **Symbolic** – логический; **Indexed** – индексированный.

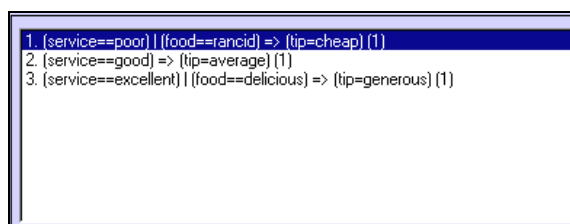


Рисунок 12 - база знаний в формате **Symbolic**.

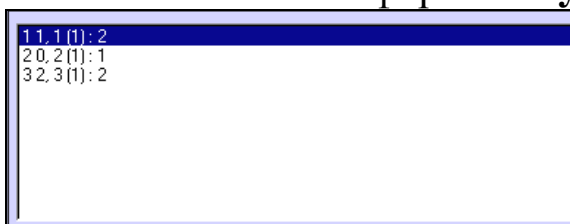


Рисунок 13 - база знаний в формате **Indexed**.

Визуализация нечеткого логического вывода

Визуализация нечеткого логического вывода осуществляется с помощью GUI-модуля **Rule Viewer**. Этот модуль позволяет проиллюстрировать ход логического вывода по каждому правилу, получение результирующего нечеткого множества и выполнение процедуры дефаззификации. **Rule Viewer** может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **View rules ...** меню **View** или нажатием клавиш **Ctrl+4**. Вид **Rule Viewer** с указанием функционального назначения основных полей графического окна приведен на рисунке 14.

Rule Viewer содержит четыре меню - **File, Edit, View, Options**, два поля ввода информации – **Input** и **Plot points** и кнопки прокрутки изображения влево-вправо (**left-right**), вверх-вниз (**up-down**). В нижней части графического окна расположены также кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Каждое правило базы знаний представляется в виде последовательности горизонтально расположенных прямоугольников. При этом первые два прямоугольника отображают функции принадлежности термов посылки правила (ЕСЛИ-часть правила), а последний третий прямоугольник соответствует функции принадлежности термина-следствия выходной переменной (ТО-часть правила). Пустой прямоугольник в визуализации второго правила означает, что в этом правиле посылка по переменной *food* отсутствует (*food is none*). Желтая заливка графиков функций принадлежности входных переменных указывает на несколько значений входов, соответствуют термам данного правила. Для вывода правила в формате **Rule Editor** необходимо сделать однократный щелчок левой кнопки мыши по номеру соответствующего правила. В этом случае указанное правило будет выведено в нижней части графического окна.

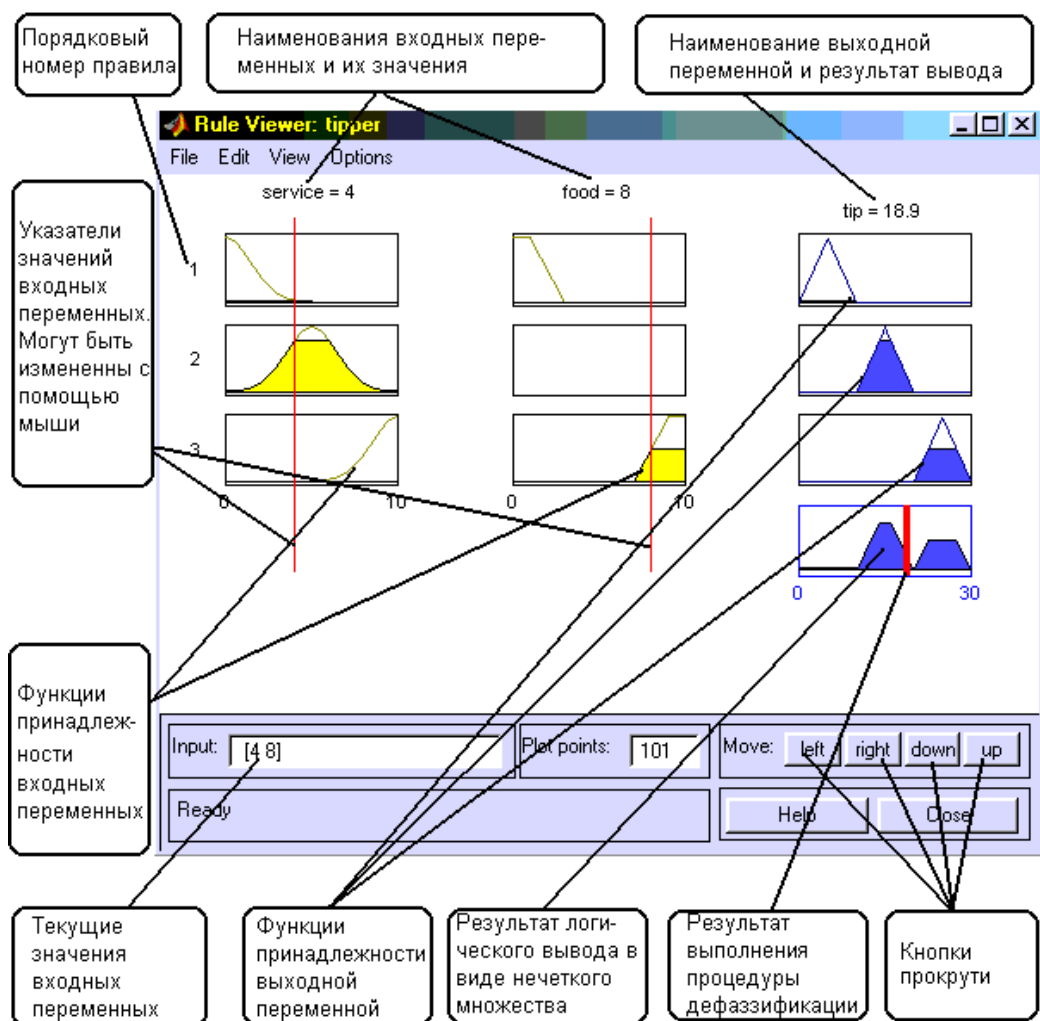


Рисунок 14 – основное меню визуализации логического вывода с помощью **Rule Viewer**

Голубая заливка графика функции принадлежности выходной переменной представляет собой результат логического вывода в виде нечеткого множества по данному правилу. Результирующее нечеткое множество, соответствующее логическому выводу по всем правилам показано в нижнем прямоугольнике последнего столбца графического окна. В этом же прямоугольнике красная вертикальная линия соответствует четкому значению логического вывода, полученного в результате дефаззификации.

Ввод значений входных переменных может осуществляться двумя способами:

- путем ввода численных значений в поле **Input**;
- с помощью мыши, путем перемещения линий-указателей красного цвета.

В последнем случае необходимо позиционировать курсор мыши на красной вертикальной линии, нажать на левую кнопку мыши и не отпустить.

кая ее переместить указатель на нужную позицию. Новое численное значения соответствующей входной переменной будет пересчитано автоматически и выведено в окно **Input**.

В поле **Plot points** задается количество точек дискретизации для построения графиков функций принадлежности.

Меню Edit

Общий вид меню приведен на рисунке 15.



Undo	Ctrl+Z
FIS Properties...	Ctrl+1
Membership Functions...	Ctrl+2
Rules...	Ctrl+3

Рисунок 15 – общий вид меню.

Команда **FIS Properties...** открывает FIS-редактор. Эта команда может быть также выполнена нажатием **Ctrl+1**. Команда **Membership Functions...** открывает редактор функций принадлежности. Эта команда может быть также выполнена нажатием **Ctrl+2**. Команда **Rules...** открывает редактор базы знаний. Эта команда может быть также выполнена нажатием **Ctrl+3**.

Меню Options

Меню **Options** содержит только одну команду **Format**, которая позволяет установить один из следующих форматов вывода выбранного правила в нижней части графического окна: **Verbose** - лингвистический; **Symbolic** – логический; **Indexed** – индексированный.

Визуализация поверхности «входы-выход»

Визуализация поверхности “входы-выход” осуществляется с помощью GUI-модуля **Surface Viewer**. Этот модуль позволяет вывести графическое изображение зависимости значения любой выходной переменной от произвольных двух (или одной) входных переменных. **Surface Viewer** может быть вызван из любого GUI-модуля, используемого с системами нечеткого логического вывода, командой **View surface ...** меню **View** или нажатием клавиш **Ctrl+4**. Общий вид модуля **Surface Viewer** с указанием функционального назначения основных полей графического окна приведен на рисунке 16.

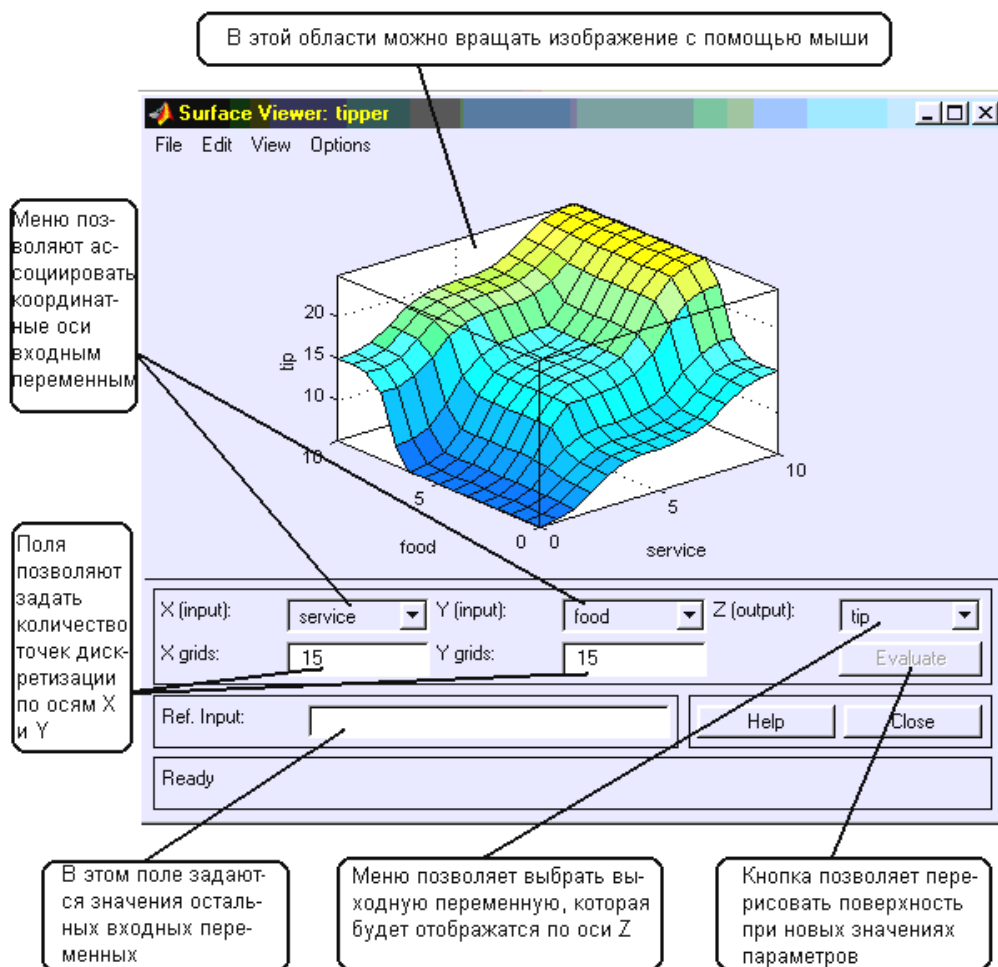


Рисунок 16 – основное меню визуализация поверхности «входы-выход» с помощью Surface Viewer.

Surface Viewer содержит верхних системных меню - **File**, **Edit**, **View**, **Options**, три меню выбора координатных осей - **X (input)**, **Y (input)**, **Z (output)**, три поля ввода информации – **X grids**, **Y grids**, **Ref. Input** и кнопку **Evaluate** для построения поверхности при новых параметрах. В нижней части графического окна расположены также кнопки **Help** и **Close**, которые позволяют вызвать окно справки и закрыть редактор, соответственно.

Surface Viewer позволяет вращать поверхность «входы-выход» с помощью мыши. Для этого необходимо позиционировать курсор мыши на поверхности «входы-выход», нажать на левую кнопку мыши и не отпуская ее повернуть графическое изображение на требуемый угол.

Поля **X grids** и **Y grids** предназначены для задания количества точек дискретизации по осям **X** и **Y**, для построения поверхности «входы-выход». По умолчанию количество дискрет по каждой оси равно 15. Для изменения этого значения необходимо установить маркер на поле **X grids** (**Y grids**) и ввести новое значение.

Поле **Ref. Input** предназначено для задания значений входных переменных, кроме тех, которые ассоциированы с координатными осями. По умолчанию это значения середины интервалов изменения переменных. Для изменения этого значения необходимо установить маркер на поле **Ref. Input** и ввести новое значение.

Меню координатных осей

Меню **X (input)**, **Y (input)**, **Z (output)** позволяют поставить в соответствие осям координат входные и выходные переменные. При этом входные переменные могут отображаться только по осям **X** и **Y**, а выходные переменные только по оси **Z**. В **Surface Viewer** предусмотрена возможность построения однофакторных зависимостей «вход-выход». Для этого в меню второй координатной оси (**X (input)** или **Y (input)**) необходимо выбрать **none**.

Меню Options

Меню **Options** изображено на рисунке 17. Оно содержит команды **Plot**, **Color Map** и **Always evaluate**.

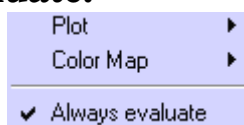


Рисунок 17 – общий вид меню.

Команда **Plot** позволяет управлять форматом вывода поверхности «входы-выход». При выборе этой команды появляется меню (рисунок 18), в котором необходимо выбрать формат вывода поверхности.

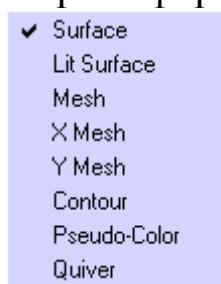


Рисунок 18 – меню команды Plot.

Команда **Color Map** позволяет управлять палитрой цветов при выводе поверхности «входы-выход». При выборе этой команды появляется меню, в котором необходимо выбрать одну из палитр: **default** – использовать палитру, установленную по умолчанию; **blue** – холодная синеголубая палитра; **hot** – теплая палитра, состоящая из черного, красного, желтого и белого цветов; **HSV** – палитра насыщенных цветов: красный, желтый, зеленый, циан, голубой, мажента, красный.

Команда **Always evaluate** позволяет установить или отменить режим автоматического, т.е. без нажатия кнопки **Evaluate**, перерисовывания поверхности «входы-выход» при любом изменении параметров.

Демонстрационный пример

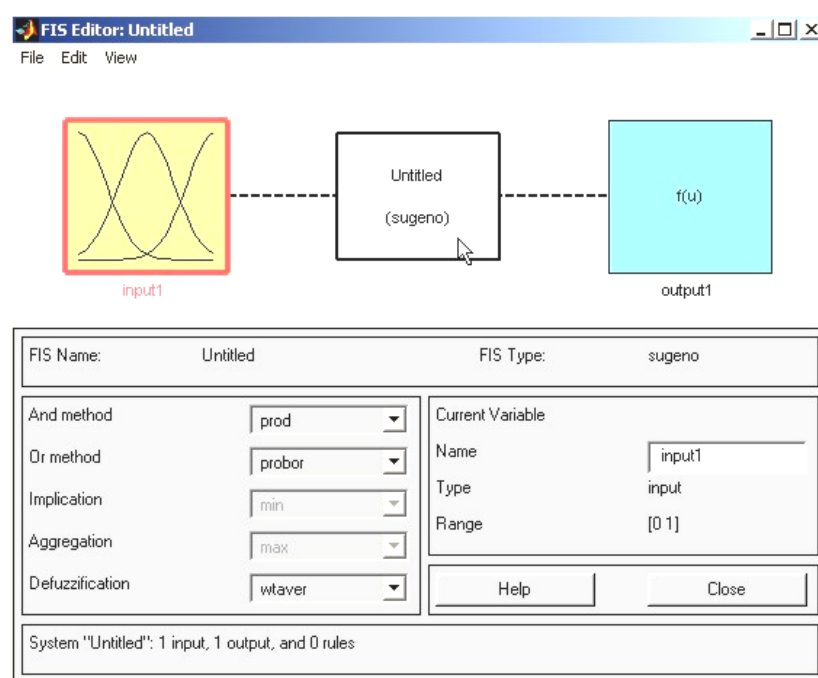
Требуется сконструировать нечеткую систему, отображающую зависимость между двумя переменными x и y . Зависимость $y = f(x)$ задана пятью точками, приведенными в таблице.

	-2	-1	0	1	2
y	-8	-1	0	1	8

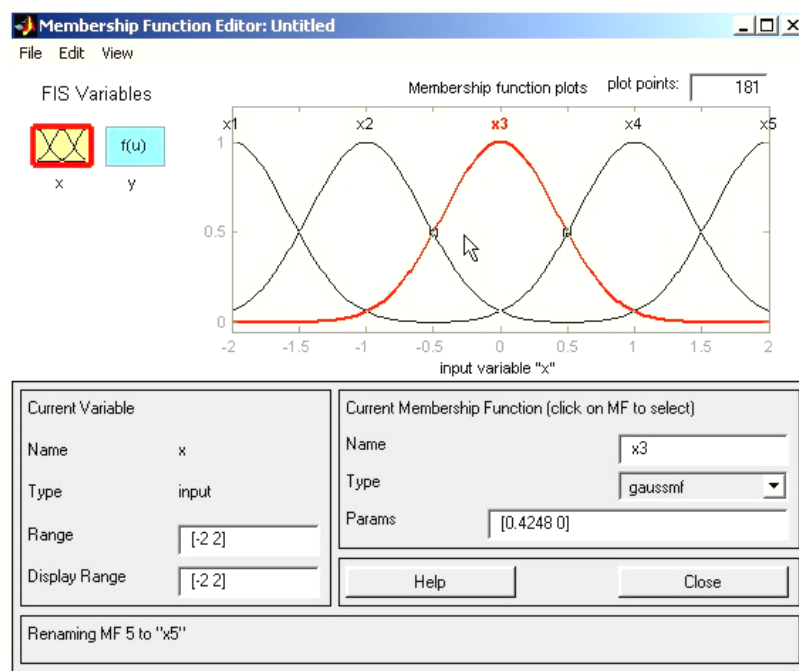
Опишем действия необходимые для построения интерполяции данной функции. Для решения задания будем использовать пакет Fuzzy Logic Toolbox (пакет нечеткой логики), входящий в состав MatLab. Этот пакет позволяет конструировать нечеткие экспертные или управляющие системы.

Откроем пункт меню Start/ Toolboxes/Fuzzy Logic/FIS Editor.

Создаем новую нечеткую систему вывода типа Sugeno: File/ New FIS /Sugeno.



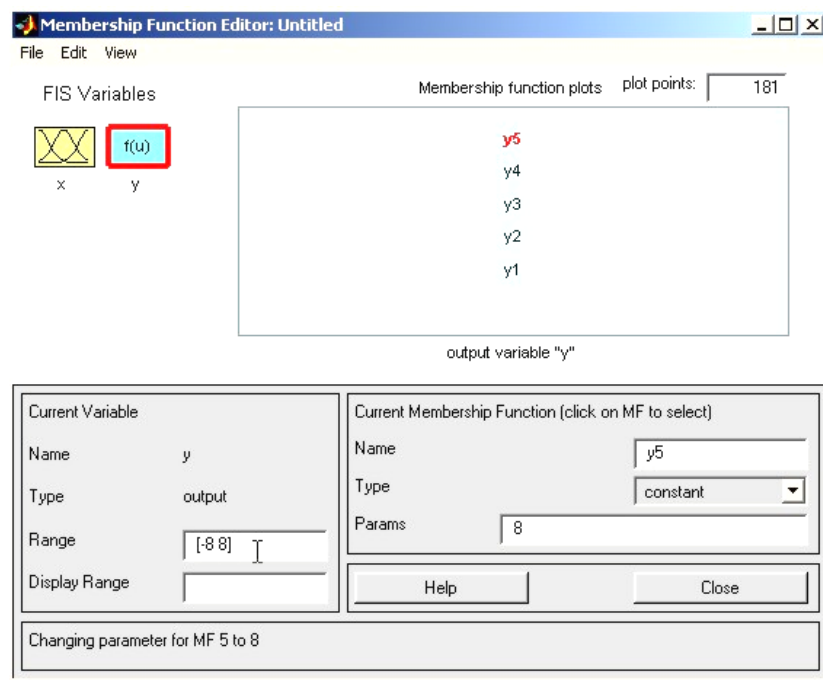
Далее в редакторе функций принадлежности добавим пять новых гауссовых функций принадлежности (для входного сигнала) при помощи Edit/ Add variables.../Input .



В поле Диапазон (Range) задаем диапазон изменения переменной x от -2 до 2, соответственно таблице.

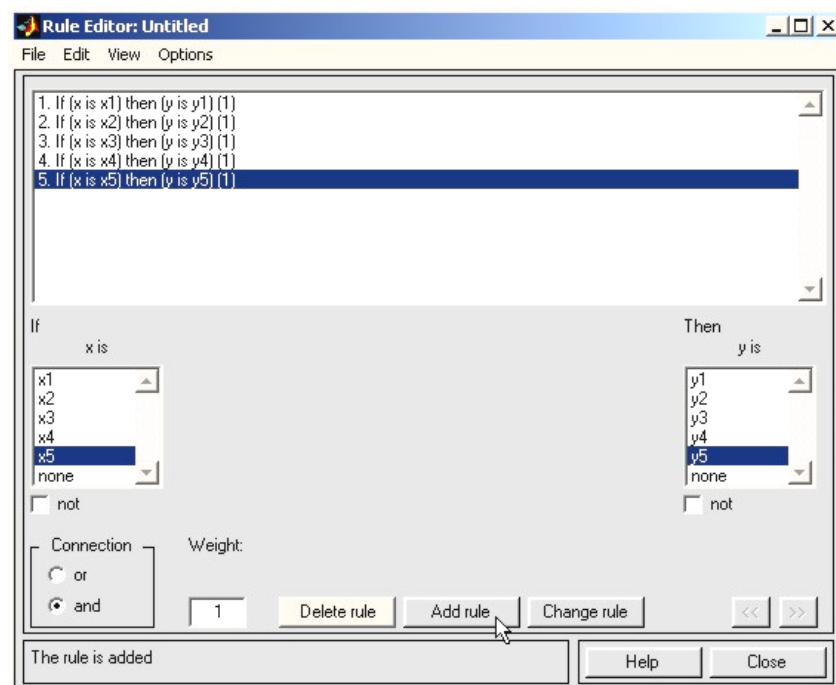
Для успешного решения поставленной задачи необходимо, чтобы ординаты максимумов функций принадлежности совпадали с заданными значениями аргумента x . Смещая графики функций принадлежности по горизонтали, добиваемся выполнения этого условия.

Зададим теперь функции принадлежности для выходной переменной y . Добавим 5 постоянных функций принадлежности (5 – число различных значений выходной переменной y , постоянные функции соответствуют алгоритму Sugeno).

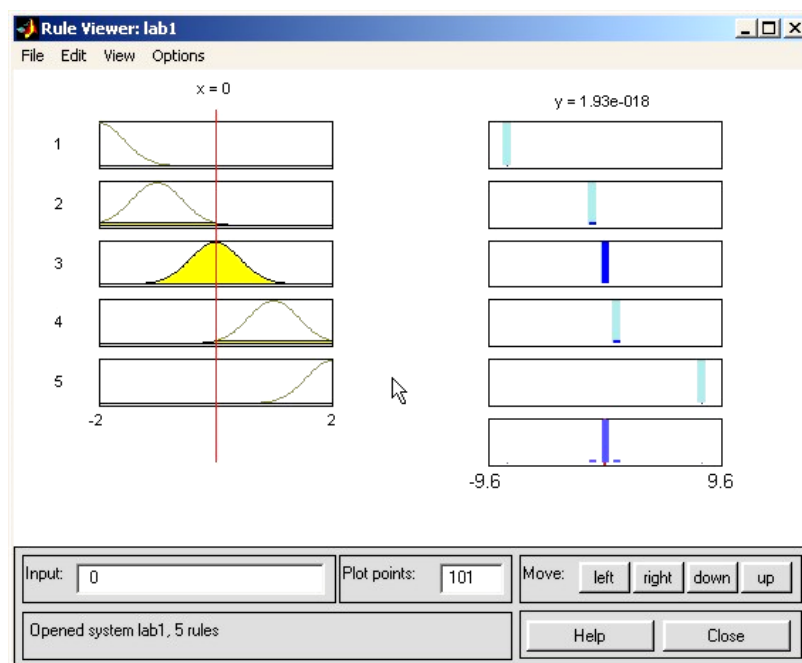


Зададим диапазон изменения выходной переменной от -8 до 8 и для каждой переменной зададим значения из таблицы.

В редакторе правил необходимо сконструировать соответствующие правила. При вводе каждого правила устанавливаем соответствие между функцией принадлежности аргумента x и числовым значением y .



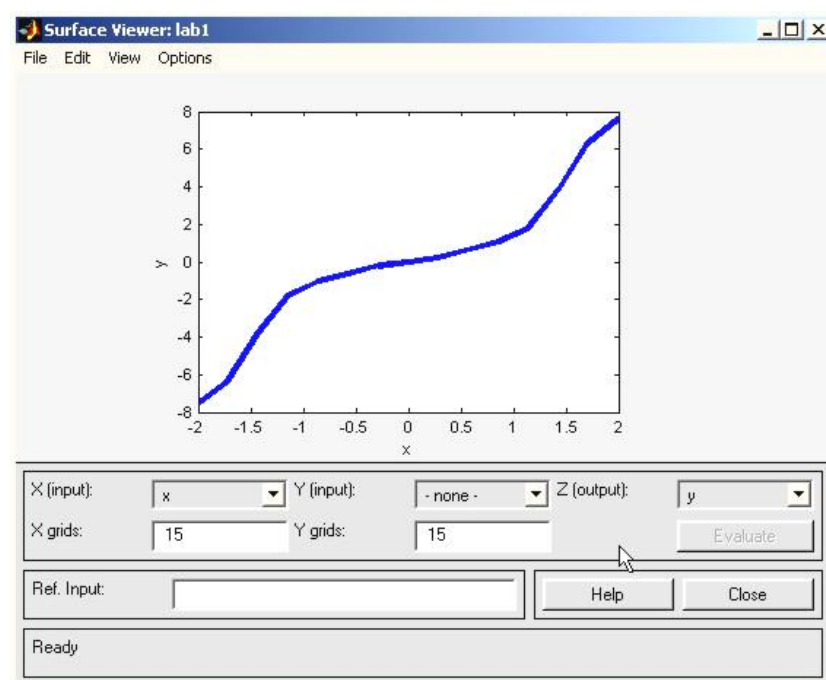
Выберем в пунктах меню Просмотр правил и изучим добавленные зависимости:



В правой части окна в графической форме представлены функции принадлежности аргумента x , а в левой – переменной выхода y с пояснением механизма принятия решений.

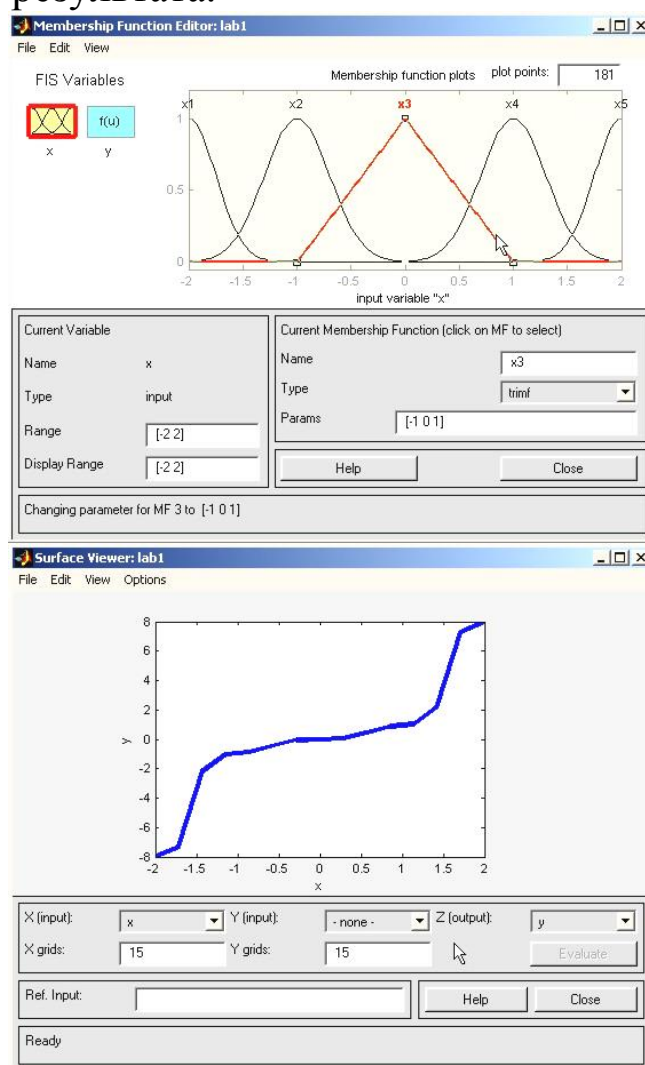
Перемещая красную черту или изменяя значения аргумента x в правой части окна, получаем значения переменной y в левой части окна. Таким образом, получено решение задачи интерполяции функции, заданной координатами пяти точек.

В окне просмотра поверхности отклика (выхода) можно визуально изучить полученный график функции $y = f(x)$.



Как видно из приведенного графика функция получилась недостаточно четкой. Это обусловлено небольшим количеством исходных точек.

Но изменяя параметры функций принадлежности можно добиться более оптимального результата.



ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Выпишите задание согласно варианта из таблицы. Рассчитайте значения двенадцати последовательных значений для заданной функции, составьте таблицу значений функции и аргумента.
2. Составить нечеткую экспертную систему, которая бы аппроксимировала функцию по таблицам значений функции и аргумента, составленной в пункте 1.
3. Смоделируйте, как изменится выходное значение при изменении выходов.

4. Рассчитайте ответ нечеткой системы «вручную» и проверьте его при помощи созданной НЭС.

Индивидуальные задания.

№ варианта	Функция f(x)
1	$\frac{x^2}{2} + 5 \cos(x)$
2	$2e^x - \frac{5}{2}x^2$
3	$e^{-x} + x^2$
4	$\frac{x^2}{(x-1)^2} + x^4$
5	$\sqrt{x+x^2} + \frac{1}{1+x^2}$
6	$\arctg(\sin(x) - \cos(x))$
7	$\frac{x^2 - 6x + 4}{3x - 2} + e^{2x}$
8	$\frac{2x^2 - 1}{\sqrt{x^2 - 2}} + x^4$
9	$sh^2(x) + e^x$
10	$\frac{1}{\ln(x)} + x^2$
11	$\frac{1}{x} + ch(x)$
12	$\frac{1}{x^2} + tg(x)$
13	$\sin(x)^2 + \sin^2(x)$
14	$\frac{1}{sh(x)} + x^4$
15	$\frac{e^{x-3}}{x-3} + \ln(x)$

Контрольные вопросы:

1. Когда рекомендуется применять нечеткие системы управления?
2. Какие наиболее часто употребляемые функции принадлежности?
3. Как реализуется механизм фузификации?
4. Как реализуется механизм дефузификации?
5. Каким образом задаются тип и параметры функций принадлежности в среде MATLAB?
6. В чем заключается метод центра максимума ?
7. В чем заключается метод наибольшего значения ?
8. В чем заключается метод центроида?
9. Каким образом осуществляется переход от нечеткого множества к четному?
10. В чем заключаются основные этапы синтеза нечеткой системы управления?

Литература:

1. Березин С.Я. Основы кибернетики и управление в биологических и медицинских системах: учебное пособие / С.Я. Березин. – Старый Оскол: ТНТ, 2013. – 244 с.
2. Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. М.: Горячая линия – Телеком, 2007.-284 с.
3. Дьяконов В.П., Круглов В.В. MATLAB 6/5 SP1/7/7 SP 5/6. Инструменты искусственного интеллекта и биоинформатика. Серия «Библиотека профессионала».-М.: СОЛОН-ПРЕСС, 2006. -456 с.
4. Емельянов, С. Г. Автоматизированные нечетко-логические системы управления [Текст]: монография / С. Г. Емельянов, В. С. Титов, М. В. Бобырь. - М.: ИНФРА-М, 2011. - 176 с. - (Научная мысль).
5. Методы робастного, нейро-нечеткого и адаптивного управления [Текст]: учебник для студентов вузов / ред. Н. Д. Егупов. - М. : МГТУ, 2001. - 743 с.