

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Кафедра систем автоматизированного проектирования
и информационных систем

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам по дисциплине **«Системы искусственного интеллекта»**
для студентов направления 09.03.02 «Информационные системы и технологии» оч-
ной формы обучения

Воронеж 2021

Составитель: канд. техн. наук Ю.В. Литвиненко

УДК 681.38+681.3

Методические указания к лабораторным работам по дисциплине «Системы искусственного интеллекта» для студентов направления 09.03.02 «Информационные системы и технологии» очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. канд. техн. наук Ю.В. Литвиненко. Воронеж, 2021. – 94 с.

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

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2003 и содержатся в файле МУ_ЛР_СИИ_ИСТЦ_2021.pdf.

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

Ответственный за выпуск зав. кафедрой

д-р техн. наук, проф. Я.Е. Львович

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

© ФГБОУ ВО «Воронежский государственный технический университет», 2021

Лабораторная работа №1

Программная реализация алгоритма формирования экспертной группы.

Методы экспертного оценивания

Цель работы: изучение подхода к формированию экспертной группы, основанный на алгоритме решения задачи о лидере. Изучение методов экспертного оценивания объектов на примере метода ранжирования. Изучение метода оценки согласованности мнений группы экспертов. Программная реализация разработанных алгоритмов.

Эффективность начальных этапов разработки экспертных систем (этапов идентификации и концептуализации) во многом определяется успешным формированием авторитетной группы экспертов и получением от них качественных знаний, составляющих основу любой экспертной системы.

При *формировании группы экспертов* на стадии выявления знаний необходимо учитывать такие характеристики экспертов как: компетентность, креативность, отношение к экспертизе, конформизм, коллективизм и самокритичность.

Рассмотрим один из возможных путей количественного описания характеристик эксперта, основанный на вычислении относительных коэффициентов компетентности по результатам высказывания специалистов о составе экспертной группы.

Суть методики сводится к тому, что ряду специалистов предлагается высказать мнение о списочном составе экспертной группы. Если в этом списке появляются лица, не вошедшие в исходный список, им тоже предлагается назвать специалистов для участия в экспертизе. После нескольких этапов будет получен достаточно полный список кандидатов в группу.

По результатам опроса составляется матрица, по строкам и столбцам которой записываются фамилии экспертов, а элементами таблицы являются переменные

При этом эксперт может включать себя или не включать в экспертную группу (то есть $x_{ij}=0$ или $x_{ij}=1$). По данной таблице можно вычислить относительные коэффициенты компетентности, используя алгоритм решения задач о лидере. Введем относительные коэффициенты компетентности h -порядка для каждого эксперта

$$K_i^h = \frac{\sum_{j=1}^m x_{ij} k_j^{h-1}}{\sum_{i=1}^m \sum_{j=1}^m x_{ij} k_j^{h-1}}, \quad (i = \overline{1, m}; h = 1, 2, \dots) \quad (1)$$

где m – число экспертов в списке (размерность матрицы $\|x_{ij}\|$), h – номер порядка коэффициента компетентности. Коэффициенты компетентности нормированы так, что их сумма равна единице:

$$\sum_{i=1}^m k_i^h = 1, \quad h = 1, 2, \dots \quad (2)$$

По формуле (1) можно вычислить значение компетентности для различных порядков, начиная с первого. При $h=1$ выражение (1) будет иметь вид:

$$k_i^1 = \frac{\sum_{j=1}^m x_{ij}}{\sum_{i=1}^m \sum_{j=1}^m x_{ij}}, \quad i = 1, 2, \dots, m \quad (3)$$

Смысл этой формулы в том, что подсчитывается число голосов, поданных за i -го эксперта и делится на общее число голосов, поданных за всех экспертов. Таким образом, коэффициент компетентности первого порядка – это относительное число экспертов, высказавшихся за включение i -го эксперта в группу.

Относительный коэффициент компетентности второго порядка получают из (1) для $h=2$ при условии, что $k_j^1 (j=1, 2, \dots, m)$ определены по (3):

$$k_i^2 = \frac{\sum_{j=1}^m x_{ij} k_j^1}{\sum_{i=1}^m \sum_{j=1}^m x_{ij} k_j^1}, \quad i = 1, 2, \dots, m \quad (4)$$

Коэффициенты второго порядка представляют собой относительное количество голосов, взвешенных коэффициентом компетентности первого порядка.

Последовательно вычисляя относительные коэффициенты компетентности более высокого порядка, можно убедиться, что процесс быстро сходится после 3-4 вычислений, то есть относительные коэффициенты быстро стабилизируются.

Допустим, что в результате опроса трех экспертов о составе экспертной группы получены данные (x_{ij}) о мнении каждого из них по включению экспертов в рабочую группу. Эти данные сведены в таблицу:

	Мнения экспертов		
	Эксперт 1 (А)	Эксперт 2 (В)	Эксперт 3 (С)
Эксперт 1 (А)	1	1	1
Эксперт 2 (В)	0	1	0
Эксперт 3 (С)	1	0	1

Результаты пошаговой обработки полученных данных по описанному выше алгоритму будут иметь вид.

На первом шаге, полагая равную компетентность всех экспертов, принимаем $k_0 = [1 \ 1 \ 1]^T$ и вычисляем коэффициенты относительной компетентности первого порядка:

$$y = \sum_{i=1}^3 \sum_{j=1}^3 x_{ij} = 1+1+1+0+1+0+1+0+1 = 6$$

$$k_A^1 = k_1^1 = \frac{1}{y} \times \sum_{j=1}^3 x_{1j} k_j^0 = \frac{1}{6} \times (1 \times 1 + 1 \times 1 + 1 \times 1) = \frac{3}{6} = 0.5$$

$$k_B^1 = k_2^1 = \frac{1}{y} \times \sum_{j=1}^3 x_{2j} k_j^0 = \frac{1}{6} \times (0 \times 1 + 1 \times 1 + 0 \times 1) = \frac{1}{6} \approx 0.167$$

$$k_C^1 = k_3^1 = \frac{1}{y} \times \sum_{j=1}^3 x_{3j} k_j^0 = \frac{1}{6} \times (1 \times 1 + 0 \times 1 + 1 \times 1) = \frac{2}{6} \approx 0.333$$

На втором шаге, используя полученные значения, вычисляем коэффициенты относительной компетентности второго порядка:

$$y = \sum_{i=1}^3 \sum_{j=1}^3 x_{ij} k_j^1 = 1 \times \frac{3}{6} + 1 \times \frac{1}{6} + 1 \times \frac{2}{6} + 0 \times \frac{3}{6} + 1 \times \frac{1}{6} + 0 \times \frac{2}{6} + 1 \times \frac{3}{6} + 0 \times \frac{1}{6} + 1 \times \frac{2}{6} = \frac{12}{6} = 2$$

$$k_A^2 = k_1^2 = \frac{1}{y} \times \sum_{j=1}^3 x_{1j} k_j^1 = \frac{1}{2} \times (1 \times \frac{3}{6} + 1 \times \frac{1}{6} + 1 \times \frac{2}{6}) = \frac{1}{2} \times \frac{6}{6} = 0.5$$

$$k_B^2 = k_2^2 = \frac{1}{y} \times \sum_{j=1}^3 x_{2j} k_j^1 = \frac{1}{2} \times (0 \times \frac{3}{6} + 1 \times \frac{1}{6} + 0 \times \frac{2}{6}) = \frac{1}{2} \times \frac{1}{6} \approx 0.083$$

$$k_C^2 = k_3^2 = \frac{1}{y} \times \sum_{j=1}^3 x_{3j} k_j^1 = \frac{1}{2} \times (1 \times \frac{3}{6} + 0 \times \frac{1}{6} + 1 \times \frac{2}{6}) = \frac{1}{2} \times \frac{5}{6} \approx 0.417$$

Продолжая аналогичные вычисления до тех пор, пока k_i^h не будут отличаться от k_i^{h-1} с точностью 0.01, получим:

При $h \rightarrow \infty \quad k^h \rightarrow [0.5 \ 0.0 \ 0.5]^T$

Существуют подходы к оценке компетентности, основанные на учете апостериорных данных, то есть результатов экспертного оценивания. Например, предложен способ оценки компетентности эксперта, базирующийся на вычислении среднего расстояния в пространстве признаков товаров, оцениваемых экспертами. Считается, что если расстояние мало, то эксперт подходит к оценке товара в целом, не различая отдельные стороны товара и, следовательно, такой эксперт недостаточно компетентен.

Суть процесса выявления знаний заключается в организации проведения экспертами интуитивно-логического анализа проблемной области с количественной оценкой формулируемых ими суждений. На этом этапе эксперты:

- формируют объекты и понятия предметной области (цели, решения, альтернативные ситуации и т.д.);
- производят измерение характеристик (вероятности свершения событий, коэффициенты значимости целей, предпочтение решений и т.д.).

Экспертное оценивание представляет собой процесс измерения, который можно определить как процедуру сравнения объектов по выбранным показателям (признакам). В этом определении фигурируют три понятия: объект, показатель (признак) и процедура сравнения.

Объектами могут быть предметы, явления, решения. В качестве показателей сравнения могут использоваться пространственно-временные, физические, психические и другие свойства и характеристики объектов. Процедура сравнения включает в себя:

- определение причинно-следственной связи между объектами;
- установление степени влияния одних объектов на другие, что требует проведения сравнения объектов, определяющих какой-либо результат, по степени их влияния на него.

К наиболее часто используемым при экспертном оценивании методам относятся: *ранжирование, парное сравнение, непосредственная оценка*. При описании

каждого из перечисленных методов полагается, что имеется конечное число измеряемых объектов и сформулирован один или несколько признаков сравнения, по которым изучается степень влияния объектов на результат.

Следовательно, методы измерения будут различаться лишь процедурой сравнения объектов. Эта процедура включает:

- построение отношений между объектами эмпирической системы;
- выбор функции f , отображающей объекты эмпирической системы на числовую систему;
- определение шкалы измерений.

Ранжирование – это процедура упорядочения объектов по степени их влияния на результат, выполняется экспертом в процессе выявления его знаний. На основе своих знаний и опыта эксперт располагает объекты в порядке предпочтения, руководствуясь одним или несколькими показателями сравнения. В зависимости от вида отношений между объектами возможны различные варианты упорядочения объектов.

Пусть среди объектов нет эквивалентных по степени влияния на результат. В этом случае между объектами существует отношение строгого порядка, обладающее свойствами:

- несимметричности (если $O_i > O_j$, то $O_i \not> O_j$);
- транзитивности (если $O_i > O_j$, $O_j > O_k$, то $O_i > O_k$);
- и связности (для любых двух объектов, либо $O_i > O_j$, либо $O_j > O_i$).

В результате сравнения всех объектов по отношению строгого порядка эксперт составляет упорядоченную последовательность:

$$O_1 > O_2 > \dots > O_n, \quad (5)$$

где объект с номером один является наиболее предпочтительным из всех объектов, объект с номером два менее предпочтителен чем первый, но предпочтительнее всех остальных и т.д.

Полученная система с отношением порядка $< O, >$ образует серию. Для серии доказано существование числовой системы:

элементами которой являются числа;

а отношение порядка $>$ есть отношение "больше чем", "предпочтительнее чем".

Это означает, что существует числовое представление $f(O_i)$, такое, что последовательности (1) соответствует последовательность чисел $f(O_1) > f(O_2) > \dots > f(O_n)$. В практике экспертного ранжирования чаще всего используется последовательность натуральных чисел $r_1=f(O_1)=1; r_2=f(O_2)=2; \dots; r_n=f(O_n)=n$.

Числа $r_1, r_2, \dots, r_n$ называются рангами. Наиболее предпочтительному присваивается ранг 1, второму – ранг 2 и т.д. На практике, среди объектов могут быть и эквивалентные по степени их влияния на результат. Например, упорядочение может иметь вид:

$$O_1 > O_2 > O_3 \sim O_4 \sim O_5 > \dots > O_{n-1} \sim O_n \quad (6)$$

В этой последовательности объекты O_3, O_4 и O_5 эквивалентны между собой, а O_{n-1} и O_n – между собой. Для эквивалентных объектов принято назначать одинаковые ранги, равные среднему арифметическому значению рангов, приписываемых одинаковым объектам. Такие ранги получили название связанных рангов. Для примера упорядочение (6) в случае $n=10$ ранги объектов O_3, O_4 и O_5 будут одинаковыми и равными:

$$r_3 = r_4 = r_5 = (3+4+5)/3 = 4$$

$$r_9 = r_{10} = (9+10)/2 = 9,5$$

Как видно из примера, связанные ранги могут быть дробными. Удобство использования связанных рангов заключается в том, что сумма рангов n -объектов равна сумме натуральных чисел от 1 до n . При этом любые комбинации связанных рангов не изменяют эту сумму. Это обстоятельство существенно упрощает обработку результатов ранжирования при групповой экспертной оценке.

При групповой экспертной оценке каждому i -ому объекту каждый из j -ых экспертов присваивает r_{ij} . В результате проведения экспертного оценивания получается

матрица рангов $\| r_{ij} \|$ размерности $n \times m$, где n – число объектов ($i=\overline{1,n}$), а m – число экспертов ($j=\overline{1,m}$).

Самый простейший способ получения *обобщенной ранжировки* заключается в ранжировании объектов по величине сумм рангов, полученных каждым объектом от всех экспертов. В этом случае для матрицы ранжировок $\| r_{ij} \|$ вычисляются суммы:

$$r_i = \sum_{j=1}^m r_{ij} \quad (i = 1, 2, \dots, n). \quad (7)$$

Далее объекты упорядочиваются по цепочке неравенств $r_k < r_l < \dots < r_q$, где $r_k = \min_i(r_i)$, $r_l = \min_{i, i \neq k}(r_i)$, $\dots$, $r_q = \max_i(r_i)$. Отсюда следует обобщенная ранжировка объектов:

$$O_k \succ O_l \succ \dots \succ O_q.$$

Для учета компетентности экспертов достаточно умножить i -ю ранжировку на коэффициенты компетентности j -го эксперта $0 \leq k_j \leq 1$. В этом случае вычисление суммы рангов для i -ого объекта производится по формуле:

$$r_i = \sum_{j=1}^m k_j r_{ij}, \quad (8)$$

что позволяет упорядочить объекты по цепочке неравенств. Следует отметить, что построение таких обобщенных ранжировок является корректной процедурой только в том случае, если ранги назначаются как места объектов в виде натуральных чисел $1, 2, \dots, n$.

Однако ранги объектов определяют только порядок расположения объектов по показателям сравнения. Ранги как числа не дают возможность сделать вывод о том, на сколько или во сколько раз предпочтительнее один объект по сравнению с другим. Если ранг 3, то отсюда не следует делать вывод о том, что объект, с рангом 1, в три раза предпочтительнее, чем объект, имеющий ранг, равный трем.

Вместе с тем для использования в ЭС знаний, полученных от экспертов, необходимо не только упорядочение или ранжирование объектов по степени их влияния или воздействия на какой-либо результат, но и определение количественной оценки степени влияния каждого из объектов на результат.

Простейшим методом для реализации этой задачи является подход, основанный на построении обобщенной ранжировки путем перехода от матрицы ранжировок к матрице парных сравнений. Для этого на основе матрицы $\|r_{ij}\|$ строится m матриц парных сравнений R_j ($j=1,2,...,m$), где m – число экспертов. Элементы этих матриц определяются следующим образом:

$$R_j = \|r_{ik}^j\| = \begin{cases} 1 & , \text{если } O_i^j \succ O_k^j, \text{ то есть } r_{ij} < r_{kj} \\ 0,5 & , \text{если } O_i^j \sim O_k^j, \text{ то есть } r_{ij} = r_{kj} \\ 0 & , \text{если } O_i^j \prec O_k^j, \text{ то есть } r_{ij} > r_{kj} \end{cases}$$

где j – номер эксперта, i и k – номера сравниваемых объектов.

Затем к полученным матрицам парных сравнений всех экспертов применяется метод обработки парных сравнений. Его итерационная процедура позволяет получить *коэффициенты относительной важности* объектов по степени их влияния на результат. Заключается он в следующем.

Задача состоит в том, что эксперт устанавливает предпочтения объектов при сравнении всех возможных пар. То есть эксперт, рассматривая все возможные пары объектов, в каждой из них устанавливает ту причину, которая, по его мнению, оказывает большее влияние на следствие. Возникает вопрос, как получить оценку всей совокупности объектов на основе результатов парного сравнения, выполненного группой экспертов.

Пусть каждый из m экспертов производит оценку влияния на результат всех пар объектов, давая числовую оценку

$$r_{ij}^h = \begin{cases} 1 & , \text{если объект } O_i \text{ более значим, чем } O_j \\ 0,5 & , \text{объекты } O_i \text{ и } O_j \text{ равноправны} \\ 0 & , \text{если объект } O_i \text{ менее значим, чем } O_j \end{cases}$$

где $h=1,2,...,m$ – номер эксперта, $i,j=1,2,...,n$ – номера объектов, исследуемых при экспертизе. Т. е. по результатам экспертизы имеем m -таблиц (матриц) вида (рис.1):

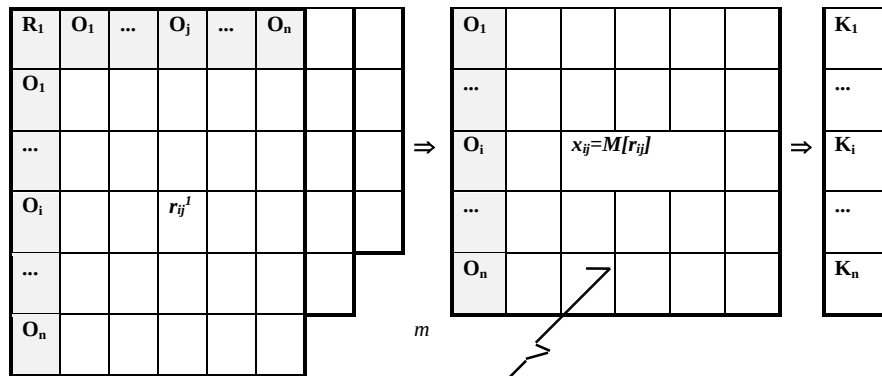


Рис.1. Последовательность обработки парных сравнений

Как следует из рис.1 последовательность обработки парных сравнений заключается в том, что на основании таблиц парных сравнений m -экспертов строится матрица математических ожиданий оценок всех пар объектов. Затем по этой матрице вычисляется вектор коэффициентов относительной важности объектов.

Если при оценке пары O_{ij} из общего количества экспертов m_i высказались в пользу предпочтения O_i , m_j экспертов в пользу O_j , а m_p считает эти объекты равноправными, то оценка математического ожидания дискретной случайной величины r_{ij} будет равна:

$$x_{ij} = M[r_{ij}^h] = 1 * \frac{m_i}{m} + 0,5 * \frac{m_p}{m} + 0 * \frac{m_j}{m}, \quad h = \overline{1, m}. \quad (9)$$

Т.к. общее количество экспертов $m = m_i + m_p + m_j$, то определяя отсюда m_p и подставляя его в вышеприведенное выражение, получим:

$$x_{ij} = \frac{m_i}{m} + 0,5 \left(\frac{m - m_i - m_j}{m} \right) = \frac{1}{2} + \frac{m_i - m_j}{2m}. \quad (10)$$

Очевидно, что $x_{ij} + x_{ji} = 1$. Совокупность величин x_{ij} образуют матрицу $X = ||x_{ij}||$ размерности $n \times n$, на основе которой можно построить ранжировку всех объектов и определить коэффициенты относительной важности объектов, то есть вектор $k = [k_1, k_2, \dots, k_n]^T$

Одним из способов определения значений элементов вектора K является итерационный алгоритм вида:

а) начальное условие $t=0$

$$k^0 = \underbrace{[1 \ 1 \ 1 \ \dots \ 1]}_n^T$$

б) рекуррентные соотношения

$$k^t = \frac{1}{\lambda^t} * X * k^{t-1}$$

$$\lambda^t = [1 \ 1 \ 1 \ \dots \ 1] * X * k^{t-1}, \quad t = (1, 2, \dots, n)$$

где X – матрица математических ожиданий оценок пар объектов, k^t – вектор коэффициентов относительной важности объектов порядка t .

$$\sum_{i=1}^n k_i^t = 1 \text{ – условие нормировки.}$$

в) признак окончания $\|k^t - k^{t-1}\| < E$.

Если матрица X неотрицательна и неразложима (то есть путем перестановки строк и столбцов ее нельзя привести к треугольному виду), то при увеличении порядка $t \rightarrow \infty$ величина λ^t сходится к максимальному собственному числу матрицы X , то есть:

$$k = \lim_{t \rightarrow \infty} k^t, \quad \sum_{i=1}^n k_i = 1.$$

Это утверждение следует из теоремы Перрона-Фробениуса и доказывает сходимость приведенного выше алгоритма.

Проиллюстрируем применение рассмотренного подхода к получению обобщенной ранжировки на примере.

Пусть три эксперта ($m=3$) провели ранжировку трех объектов ($n=3$) по степени их влияния на какой-либо результат и таблица ранжировок имеет вид:

Объект O_i	Эксперт 1	Эксперт 2	Эксперт 3
O_1	1	1	1
O_2	2	1	2
O_3	3	1	1

На основе этой таблицы матрица парных сравнений для всех экспертов будет иметь вид:

Эксперт 1(R_1) Эксперт 2(R_2) Эксперт 3(R_3)

	O_1	O_2	O_3
O_1	0,5	1	1
O_2	0	0,5	0
O_3	0	1	0,5

	O_1	O_2	O_3
O_1	0,5	0,5	0,5
O_2	0,5	0,5	0,5
O_3	0,5	0,5	0,5

	O_1	O_2	O_3
O_1	0,5	1	0,5
O_2	0	0,5	0
O_3	0,5	1	0,5

Для получения групповой оценки степени влияния каждого из объектов на результат, построим матрицу математических ожиданий оценок каждой из пар объектов, которая для рассматриваемого примера будет иметь вид:

	O_1	O_2	O_3
O_1	3/6	5/6	4/6
O_2	1/6	3/6	1/6
O_3	2/6	5/6	3/6

Значения элементов этой матрицы получены из следующих выражений:

$$x_{11} = \frac{1}{2} + \frac{0-0}{2*3} = \frac{1}{2} \quad x_{11} = x_{22} = x_{33} = \frac{3}{6}, \quad x_{12} = \frac{1}{2} + \frac{2-0}{2*3} = \frac{5}{6} \quad x_{21} = 1 - x_{12} = \frac{1}{6}$$

$$x_{13} = \frac{1}{2} + \frac{1-0}{2*3} = \frac{4}{6} \quad x_{31} = 1 - x_{13} = \frac{2}{6}, \quad x_{23} = \frac{1}{2} + \frac{0-2}{2*3} = \frac{1}{6} \quad x_{32} = 1 - x_{23} = \frac{5}{6}$$

Воспользуемся вышеописанным алгоритмом для получения вектора относительной важности объектов. Для наглядности, каждый из шагов представим в виде:

шаг 0:

$$K^0 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$$

шаг 1:

$$Y^1 = X \times K^0 = \frac{1}{6} \times \begin{bmatrix} 3 & 5 & 4 \\ 1 & 3 & 1 \\ 2 & 5 & 3 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{6} \times \begin{bmatrix} 3+5+4 \\ 1+3+1 \\ 2+5+3 \end{bmatrix} = \frac{1}{6} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix}$$

$$\lambda^1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times Y^1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times \frac{1}{6} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \frac{1}{6} \times 27 = \frac{27}{6}$$

$$K^1 = \frac{1}{\lambda^1} \times Y^1 = \frac{6}{27} \times \frac{1}{6} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \frac{1}{27} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \begin{bmatrix} 0,444 \\ 0,185 \\ 0,370 \end{bmatrix}$$

шаг 2:

$$Y^2 = X \times K^1 = \frac{1}{6} \times \begin{bmatrix} 3 & 5 & 4 \\ 1 & 3 & 1 \\ 2 & 5 & 3 \end{bmatrix} \times \frac{1}{27} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \frac{1}{6 \times 27} \times \begin{bmatrix} 36 + 25 + 40 \\ 12 + 15 + 10 \\ 24 + 25 + 30 \end{bmatrix} = \frac{1}{6 \times 27} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix}$$

$$\lambda^2 = [1 \ 1 \ 1] \times Y^2 = [1 \ 1 \ 1] \times \frac{1}{6 \times 27} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix} = \frac{217}{6 \times 27}$$

$$K^2 = \frac{1}{\lambda^2} \times Y^2 = \frac{6 \times 27}{217} \times \frac{1}{6 \times 27} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix} = \frac{1}{217} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix} = \begin{bmatrix} 0,465 \\ 0,171 \\ 0,364 \end{bmatrix}$$

$$\max (|0,465 - 0,444| , |0,171 - 0,185| , |0,364 - 0,3701|) = 0,021 > 0,001$$

Продолжая итерационный процесс до тех пор, пока норма оценки не будет меньше заданной ($\max_i (|K_i^t - K_i^{t-1}|) < 0,001$) получим:

$$K^3 = [0,468 \ 0,169 \ 0,363]^T$$

$$K^4 = [0,468 \ 0,169 \ 0,363]^T$$

На четвертом шаге выполняется условие выхода, что позволяет за групповую оценку степени влияния на результат принять вектор коэффициентов относительной важности объектов вида:

$$K = [0,468 \ 0,169 \ 0,363]^T,$$

что позволяет оценить количественно степень влияния каждого объекта на результат, полученный на основе исходного ранжирования экспертов.

Все существующие методы получения групповых оценок позволяют получить достоверные результаты в случае хорошо подобранной группы экспертов и согласованности их мнений. Определение согласованности мнений экспертов производится путем вычисления числовой меры, характеризующей степень близости индивидуальных мнений. Анализ значения меры согласования способствует выработке правильного суждения об общем уровне знаний по решаемой проблеме и выявлению группировок мнений экспертов.

Существует много подходов к решению этой проблемы. Рассмотрим способ оценки согласованности по данным матрицы ранжирования с использованием коэффициента конкордации:

$$W = \frac{S(d^2)}{\frac{1}{12} m^2 (k^3 - 1) - m \sum_{j=1}^m T_j}, \quad (11)$$

где $j = \overline{1, m}$ (m – количество экспертов), k – количество объектов, $S(d^2)$ – сумма квадратов разностей:

$$d = \left(\sum_{j=1}^m a_{ij} \right) - \frac{1}{2} m(k+1), \quad (12)$$

где a_{ij} - обобщенная сумма рангов i -ого объекта.

$$T_j = \frac{1}{12} \sum_{i=1}^k (t_j^3 - t_j), \quad (13)$$

где t_j - число повторений i -ого ранга в j -ой строке матрицы.

Если исходная матрица ранжирования не содержит совпавших рангов, то:

$$W = \frac{12 \cdot S(d^2)}{m^2 (k^3 - 1)} \quad (14)$$

Оценка значимости коэффициента конкордации производится по χ^2 -критерию Пирсона:

$$\chi^2_{\delta} = m(k-1)W \quad (15)$$

Если при числе степеней свободы $f=k-1$ критическое значение χ^2_{δ} окажется меньше χ^2_{δ} , то гипотеза о наличии согласованности мнений экспертов принимается.

Например, пусть восемь экспертов оценили пять объектов по степени влияния на результат. Данные занесены в матрицу ранжирования. Требуется проверить гипотезу о наличии согласованности мнений экспертов.

k=5	Эксперты m=8								a_{ij}	d	d_i^2
1	1	3	2	2	2	2	1	1	14	-10	100
2	2	4	3	5	3	4	2	2	25	1	1
3	3	2	4	1	4	3	5	5	27	3	9
4	4	5	5	3	5	5	4	4	35	11	121
5	5	1	1	4	1	1	3	3	19	-5	25
$S(d^2)=256$											

$$W = \frac{12 \cdot 256}{8^2 \cdot (5^3 - 1)} = 0,384$$

$$\chi^2_{\delta} = 12,288$$

$\chi^2_{\text{ед}} = 9,488$ при числе степеней свободы 4 и уровне значимости $Q=5\%$.

Так как $\chi^2_{\text{д}}$ больше $\chi^2_{\text{ед}}$, то гипотеза о наличии согласованности мнений экспертов принимается.

Задание на лабораторную работу

1. В рамках своей подгруппы определить некоторую проблемную область. Произвести опрос по поводу включения членов подгруппы в рабочую группу экспертов в выбранной проблемной области. По результатам опроса сформировать матрицу.

2. Разработать программу, осуществляющую обработку результатов опроса по алгоритму решения задач о лидере, на основании чего принимается решение о составе экспертной группы.

3. Для выбранной проблемной области определить ряд объектов, оказывающих влияние на результат. Каждый из членов экспертной группы должен проранжировать эти объекты. Результаты представить в виде матрицы ранжирования.

4. Разработать программу, осуществляющую вычисление обобщенных ранжировок. Результатом работы программы является вектор-столбец коэффициентов относительной важности объектов.

5. Оценить согласованность мнений группы экспертов на основании данных матрицы ранжирования с использованием коэффициента конкордации.

6. Оформить отчет. Отчет должен содержать:

1. Теоретический минимум.
2. Матрицу результатов опроса о составе экспертной группы и результаты обработки.
3. Матрицу ранжирования объектов.
4. Вектор-столбец коэффициентов относительной важности объектов.
5. Оценку согласованности мнений группы экспертов.
6. Листинги программ.

Лабораторная работа №2

Байесовские сети доверия

Целью работы является изучение основ проектирования экспертных систем на основе байесовских сетей доверия, приобретение навыков работы в системе Hugin.

В ходе выполнения лабораторной работы будет рассмотрен один из подходов к реализации баз знаний экспертных систем и осуществлена практическая реализация экспертной системы в системе Hugin. Для выполнения лабораторной работы необходимо иметь доступ к компьютеру с установленной системой Hugin.

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

- знание структуры и механизма работы байесовских сетей доверия;
- знание назначения и структуры экспертных систем;
- знание основных подходов к построению экспертных систем;
- знание назначения и принципов работы в системе Hugin;
- навыки построения базы знаний предметной области;
- навыки построения байесовских сетей доверия с помощью системы Hugin.

Плохо поставленные (некорректные) задачи существуют во всех предметных областях, где могут быть эффективны интеллектуальные системы. В области некорректных задач точные знания нельзя получить, либо их нельзя поучить сразу, поэтому, как правило, необходим подход, суть которого в постепенном приближении к полному набору знаний. Другими словами, знания чаще всего нечетки. Для того чтобы интеллектуальные системы вышли за рамки простых символьных выводов и приблизились к мышлению человека, необходимы методы представления нечетких знаний и механизм выводов, работающий в их среде. В настоящее время модели представления знаний являются предметом исследований и разработок в среде узких специалистов, в основном программистов и математиков, в то время как потребность в таких моделях ощущается практически во всех предметных областях. Это объясняется необходимостью построения экспертных систем, структура которых неразрывно связана с формами представления знаний, определяемых в свою очередь

особенностями данной предметной области. Особенность экспертных систем заключается в том, что они моделируют деятельность человека, осуществляемую часто в неформальном виде. Знания не всегда могут быть описаны точно — часто встречаются так называемые "нечеткие" знания. Люди повседневно решают проблемы и делают заключения в среде нечетких знаний, и для того, чтобы интеллектуальные системы обладали такими возможностями, как гибкость, широкий кругозор, адаптируемость, необходимо представление и использование нечетких знаний. При обработке нечетких знаний возникает очевидная необходимость иметь количественную оценку для такого рода высказываний. Такая количественная оценка носит название нечеткой меры (или меры неопределенности). Принципы обработки нечетких знаний основаны на логике. Основное преимущество вероятностного рассуждения над логическим — в разрешении фактора достижения равно рациональных решений даже при отсутствии достаточного объема информации. Совместное распределение вероятностей может ответить на любой вопрос о предметной области, но может оказаться сложным, поскольку число переменных растет. В контексте использования формул Байеса, установление связей между условно независимыми переменными могут упростить вычисление результатов запроса и существенно уменьшить количество условных вероятностей, которые должны быть определены.

Для представления зависимости между переменными и краткой спецификации совместного распределения вероятностей используют структуру данных, называемую байесовской сетью (сетью доверия и т.п.). Рассмотрим структуру и механизм работы байесовских сетей доверия. Байесовская сеть доверия (БСД) представляет собой направленный ациклический граф, дуги которого представляют собой причинно - следственные связи, направленные от причин к следствиям, а узлы представляют собой состояния переменных, характеризующих конкретную задачу. Базовым положением БСД является гипотеза о том, что человек — эксперт способен формулировать правила, связывающие не более двух — трех элементарных утверждений. Таким образом, база знаний должна состоять из частей, соответствующих этим правилам. Между такими частями базы знаний также существуют определенные взаимоотношения. Такое разбиение базы правил представлено графом БСД,

например, рис.1. Исходным данным соответствуют начальные узлы БСД, а результаты работы сети определяются на конечных узлах сети, представляющих переменные, характеризующие конкретную задачу.

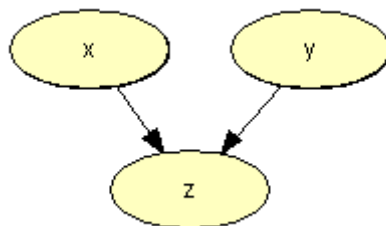


Рис.1. Пример простейшей БСД

На рис.1 видно, что вершины x и y не имеют предков и являются родительскими по отношению к вершине z , которая, соответственно, является дочерней по отношению к ним. Если вершина в БСД может находиться в одном из нескольких состояний (а не принимать любое значение из некоторого диапазона), то она является дискретной. Для таких вершин в качестве меры истинности в аппарате БСД используется мера вероятности. Связь между переменными, участвующими в правиле, передается с помощью тензора условной вероятности (в данном примере $P(z/x,y)$). Состояния вершин, не имеющих предков, описываются безусловными вероятностями, например, $P(x), P(y)$.

Топологию сети можно рассматривать как абстрактную базу знаний, которая содержит целый ряд установочных параметров, в виду того, что она представляет общую структуру причинных процессов в области, а не специфические детали, байесовская сеть позволяет полностью описать область. Каждый элемент, входящий в совместное распределение вероятностей, может быть вычислен, исходя из информации в сети. Вхождение объединенных в единое целое элементов — это вероятность конъюнкции конкретных значений переменных, т.е., каждое такое вхождение представляется функцией соответствующих элементов таблиц условных вероятностей байесовской сети. Значит, таблицы условных вероятностей обеспечивают разложение целого на составные части. Поскольку каждый узел соединен только с узлами, расположенными выше, то технология построения сети гарантирует, что сеть является нециклической. Другое важное свойство байесовских сетей заключается в том, что они не содержат никакие избыточные значения вероятности,

за исключением, возможно, одного вхождения в каждую строку каждой таблицы условных вероятностей. Это означает, что это ни специалист по технике представления и использования знаний, ни эксперт не могут создать сеть доверия, которая бы нарушала аксиомы вероятности.

Рассмотрим пример построения простейшей экспертной системы с помощью байесовской сети доверия. Пусть проблемная область описывается следующей совокупностью правил.

П1: ЕСЛИ компьютер упал, ТО на жестком диске появляются дефектные сектора ИЛИ происходит нарушение контактов блока питания.

П2: ЕСЛИ на жестком диске появляются дефектные сектора, ТО компьютер не загружается ИЛИ при диагностике логический диск недоступен.

П3: ЕСЛИ происходит нарушение контактов блока питания, ТО компьютер не загружается ИЛИ индикатор питания не горит.

П4: ЕСЛИ компьютер не загружается, ТО нет изображения на экране монитора.

Данная проблемная область может быть смоделирована байесовской сетью доверия, содержащей семь вершин событий, каждая из которых может пребывать в одном из двух состояний:

- вершина t - «падение компьютера со стола»;
- вершина u - «появление на жестком диске дефектных секторов»;
- вершина v - «нарушение контактов в блоке питания»;
- вершина w - «загрузка компьютера»;
- вершина x - «доступность логического диска»;
- вершина y - «наличие изображения на экране монитора»;

- вершина z - «состояние индикатора питания».

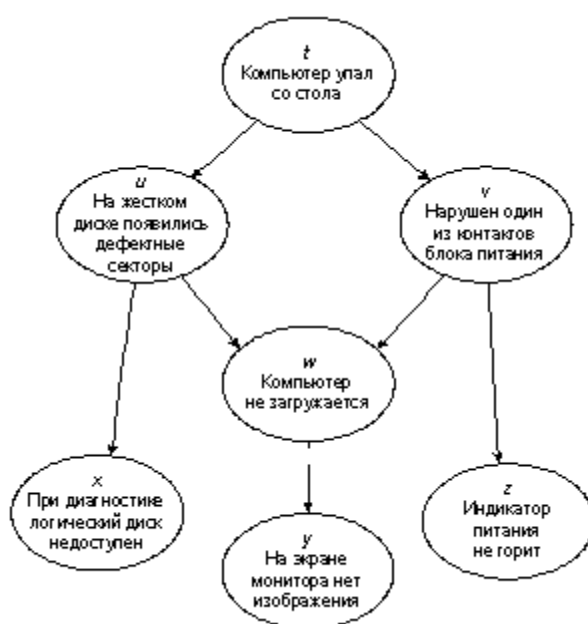


Рис.2. Байесовская сеть доверия

Байесовская сеть доверия, представленная на рис.2, моделирует причинно-следственные связи между вершинами в соответствии с правилами, описывающими проблемную область.

Однако, это только качественное представление. Перед тем, как назвать это полностью байесовской сетью доверия, необходимо определить количественное представление, то есть множество таблиц безусловных и условных вероятностей. В данной БСД только вершина t не имеет родителей, следовательно, она описывается таблицей безусловных вероятностей (табл.1), а остальные вершины – таблицами условных вероятностей (табл.2 – табл.7).

Таблица 1 - Задание вершины t

Вершина t - «падение компьютера со стола»	
$p(t)$, т.е. $P(\text{«Компьютер упал со стола»})$	$p(\bar{t})$, т.е. $P(\text{«Компьютер не падал со стола»})$
0,001	0,999

Таблица 2 - Задание вершины u

Вершина u - «появление на жестком диске дефектных секторов»		
	$t = \text{«Компьютер упал со стола»}$	$\bar{t} = \text{«Компьютер не падал со стола»}$
$u = \text{«на жестком диске появились дефектные секторы»}$	$p(u/t) = 0,16$, т.е. $P(\text{«на жестком диске появились дефектные секторы, если компьютер упал со стола»})$	$p(u/\bar{t}) = 0,05$, т.е. $P(\text{«на жестком диске появились дефектные секторы, если компьютер не падал со стола»})$
$\bar{u} = \text{«на жестком диске не появились дефектные секторы»}$	$p(\bar{u}/t) = 0,84$, т.е. $P(\text{«на жестком диске не появились дефектные секторы, если компьютер упал со стола»})$	$p(\bar{u}/\bar{t}) = 0,95$, т.е. $P(\text{«на жестком диске не появились дефектные секторы, если компьютер не падал со стола»})$

Таблица 3 - Задание вершины v

Вершина v - «нарушение контактов в блоке питания»		
	$t = \text{«Компьютер упал со стола»}$	$\bar{t} = \text{«Компьютер не падал со стола»}$
$v = \text{«нарушен один из контактов блока питания»}$	$p(v/t) = 0,1$, т.е. $P(\text{«нарушен один из контактов блока питания, если компьютер упал со стола»})$	$p(v/\bar{t}) = 0,001$, т.е. $P(\text{«нарушен один из контактов блока питания, если компьютер не падал со стола»})$
$\bar{v} = \text{«не нарушены контакты блока питания»}$	$p(\bar{v}/t) = 0,9$, т.е. $P(\text{«не нарушены контакты блока питания, если»})$	$p(\bar{v}/\bar{t}) = 0,999$, т.е. $P(\text{«не нарушены контакты блока питания, если»})$

	компьютер упал со стола»)	если компьютер не па- дал со стола»)
--	------------------------------	---

Таблица 4 - Задание вершины w

Вершина w - «загрузка компьютера»				
	u = «на жестком диске появились дефектные секторы»		$\bar{u}$ = «на жестком диске не появились дефектные секторы»	
	v= «нарушен один из контактов блока питания»	$\bar{v}$ = «не наруше ны контакты блока питания»	v= «нарушен один из контактов блока питания»	$\bar{v}$ = «не наруше ны контакты блока питания»
w= «компьютер не загружается»	$p(w/uv)=0,95$	$p(w/u\bar{v})=0,25$	$p(w/\bar{u}v)=0,9$	$p(w/\bar{u}\bar{v})=0,02$
$\bar{w}$ = «компьютер загружается»	$p(\bar{w}/uv)=0,05$	$p(\bar{w}/u\bar{v})=0,75$	$p(\bar{w}/\bar{u}v)=0,1$	$p(\bar{w}/\bar{u}\bar{v})=0,98$

Таблица 5 - Задание вершины x

Вершина x - «доступность логического диска»		
	u= «на жестком диске появились дефектные секторы»	$\bar{u}$ =«на жестком диске не появились дефектные секторы»
x= «при диагностике ло- гический диск недосу- пен»	$p(x/u)= 0,2$	$p(x/\bar{u})= 0,025$
$\bar{x}$ = «при диагностике логический диск досту- пен»	$p(\bar{x}/u)= 0,8$	$p(\bar{x}/\bar{u})= 0,975$

Таблица 6 - Задание вершины у

Вершина у - «наличие изображения на экране монитора»		
	w= «на жестком диске появились дефектные секторы»	$\bar{w}$ =«на жестком диске не появились дефектные секторы »
y= « нет изображения на экране монитора»	$p(y/w) = 0,4$	$p(y/\bar{w}) = 0,001$
$\bar{y}$ = « есть изображение на экране монитора»	$p(\bar{y}/w) = 0,6$	$p(\bar{y}/\bar{w}) = 0,999$

Таблица 7 - Задание вершины z

Вершина z - «состояние индикатора питания»		
	v= «нарушен один из контактов блока питания»	$\bar{v}$ =« не нарушены контакты блока питания»
z= «индикатор питания не горит»	$p(z/v) = 0,99$	$p(z/\bar{v}) = 0,001$
$\bar{z}$ = «индикатор питания горит»	$p(\bar{z}/v) = 0,6$	$p(\bar{z}/\bar{v}) = 0,999$

Современные программные средства (такие как MSBN, HUGIN и др.) обеспечивают инструментарий для построения БСД, а также возможность использования БСД для введения новых свидетельств и получения решения (вывода) за счёт пересчёта вероятностей во всех вершинах, соответствующих этим свидетельствам.

При реализации БСД в среде HUGIN следует иметь в виду, что данная система не во всех режимах поддерживает работу с русскими шрифтами. Поэтому для обозначения вершин будем пользоваться английскими шрифтами.

После запуска HUGIN Runtime, откроется окно системы HUGIN (рис.3), которое содержит панель меню, панель инструментов (tool bar pane), а также панель редактирования вершин (node edit pane) и панель графического отображения сети (network pane). Новая пустая сеть доверия получает имя "unnamed1" и автоматически открывается в окне сети (network window).

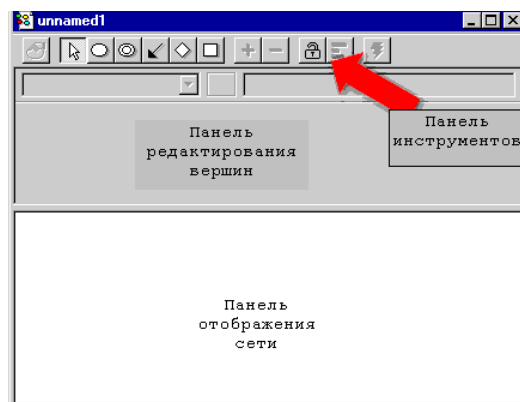


Рис.3. Окно системы HUGIN

После запуска системы HUGIN (рис. 3) она автоматически устанавливается в режим редактирования (edit mode), что позволяет немедленно приступить к построению новой БСД и определению состояния ее вершин.

Другим важным режимом работы системы является режим выполнения (run mode), который позволяет использовать БСД для получения требуемых результатов.

Построение БСД начинается с определения отдельных вершин, входящих в проектируемую БСД. При этом для создания вершин с дискретными состояниями, определения их свойств и задания причинно - следственных связей между вершинами используются приведенные на рис.4 инструменты интерфейса системы, которые активируются путем нажатия левой кнопки мыши на их пиктограммах.



- вершины с дискретными состояниями [discrete chance tool]



- свойства вершин [node properties tool]



- добавление связей [casual arrow tool]

Рис.4. Инструменты интерфейса системы

Для создания вершины БСД необходимо установить режим добавления вершин с дискретными состояниями, выбрав соответствующую пиктограмму в панели

инструментов (рис. 4) и щелкнуть мышью в любом месте панели отображения сети (рис. 3), где предполагается размещение добавляемой вершины. После этого в панели отображения сети должна появиться в виде овала вершина, название которой по умолчанию будет C1, C2,...,Cn.

Далее необходимо присвоить вершине имя в соответствии с моделью. Для этого необходимо выделить курсором мыши нужную вершину, установить режим определения свойств вершины [Node Properties], щелкнув на соответствующей пиктограмме панели инструментов (рис.4) и изменить содержимое полей Name и Label на требуемое. Name - это внутреннее имя вершины, пока Label является названием вершины. Если поле Label не задано (как это было до того момента, пока мы не изменили поле Label), то его значение соответствует внутреннему имени. Внутреннее имя может содержать символы 'a' - 'z' и 'A' - 'Z', цифры '0' - '9', а также знак подчеркивания '_'.

Можно добавлять вершины, не нажимая каждый раз на пиктограмму вершин с дискретными состояниями, а, однажды выбрав режим добавления и удерживая клавишу SHIFT, щелкнуть мышью в окне отображения сети столько раз, сколько вершин с необходимо добавить в БСД. Результат этих действий представлен на рис.5.

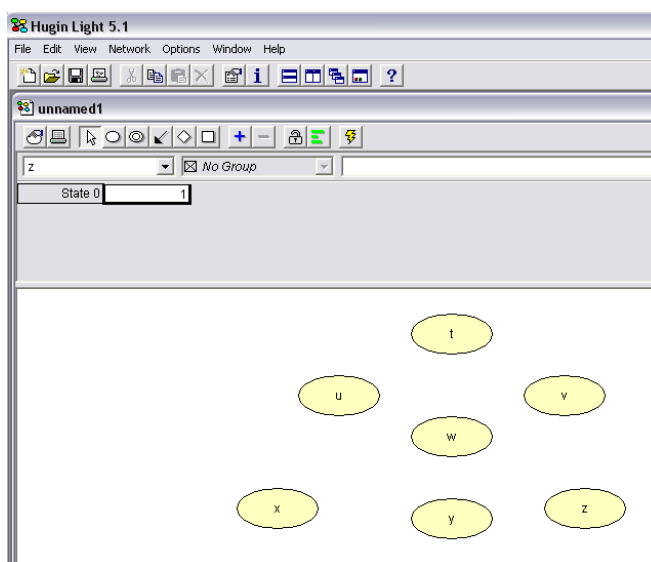


Рис.5. Добавление вершин в БСД

На следующем этапе проектирования БСД необходимо установить причинно-следственные связи между вершинами. Эти связи в модели БСД отображаются в виде стрелок, соединяющих между собой вершины. Необходимо нажать иконку добавления связей (рис. 4) и протянуть мышью стрелку от родительской к дочерней вершине, нажав левую кнопку. Результат этих действий представлен на рис.6.

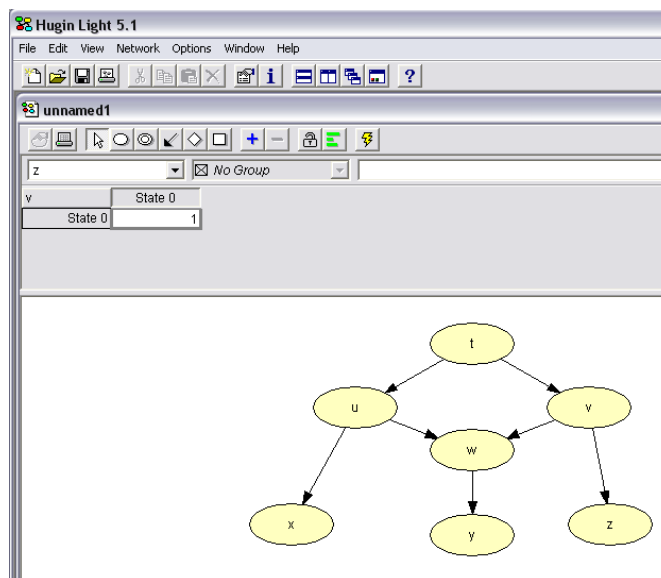


Рис.6. Добавление связей в БСД

Следующий шаг - это задание состояний и таблиц безусловных и условных вероятностей для каждой вершины. Ранее все вершины были определены так, что каждая из них может находиться в одном из двух состояний. Рассмотрим, как задаются эти состояния в системе HUGIN. Для этого необходимо сделать активной вершину, выбрав ее в раскрывающемся списке на панели инструментов или дважды щелкнув на ней мышью, нажать пиктограмму добавления состояний на панели инструментов («+» - добавить состояние, «-» - удалить состояние). Далее нужно, щелкнув мышью, перевести курсор в таблицу условных вероятностей на поле, содержащем текст State 0 и ввести в него название первого состояния. Затем State 1 заменить на название второго состояния вершины. Следует заметить, что таблицы для вершин без родителей и для вершин с одним или несколькими родителями существенно отличаются (рис.7, 8, 9).

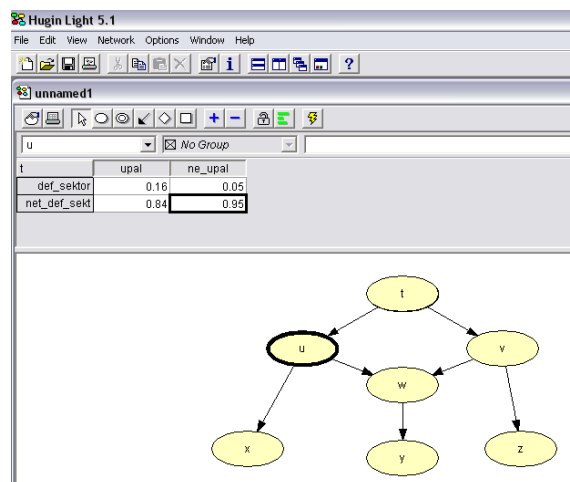


Рис.7. Задание таблицы безусловных вероятностей для вершины без родителей

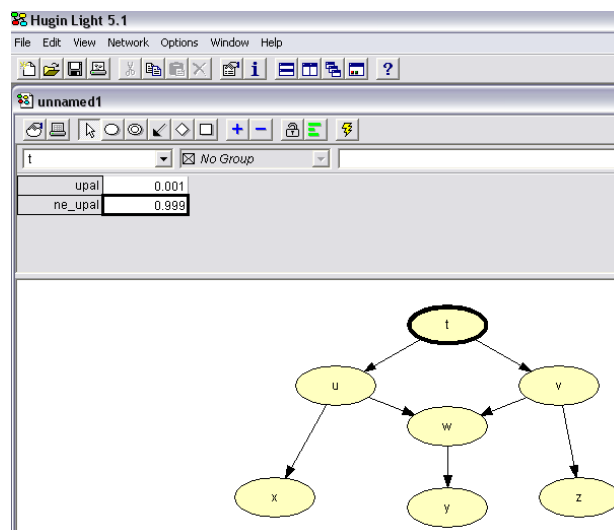


Рис.8. Задание таблицы условных вероятностей для вершины с одним родителем

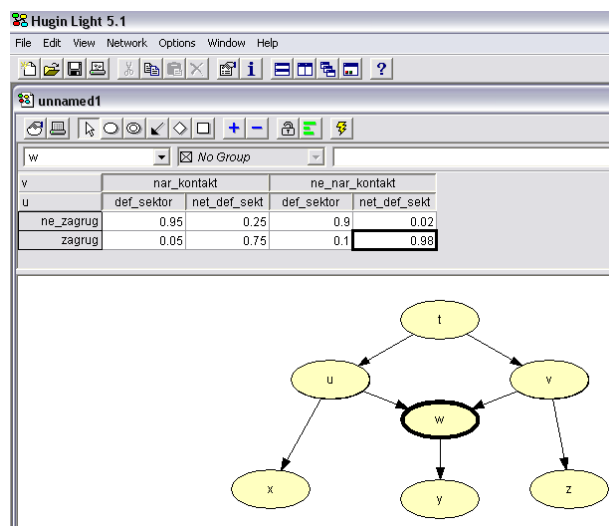


Рис. 9. Задание таблицы условных вероятностей для вершины с несколькими родителями

После того, как БСД спроектирована и определена в системе HUGIN, ее необходимо прокомпилировать и запустить (перевести в режим вычислений). Для этого нужно щелкнуть мышкой по пиктограмме режима вычислений [run mode tool button] в панели инструментов (рис.10).



Рис.10. Пиктограмма режима вычислений

При правильном выполнении предыдущих шагов работы, компиляция закончится успешно и быстро, и система перейдет в режим вычислений ["run" mode] (рис. 11).

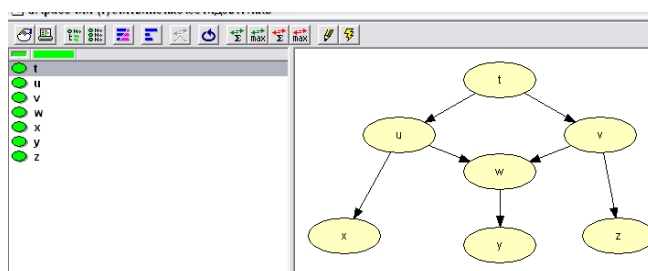


Рис.11. Окно программы в режиме вычислений

В этом режиме окно сети поделено на две вертикальные секции. Левая секция представляет собой панель списка вершин [node list pane], а правая - панель отображения сети [network pane]. В левой части окна отображаются вершины БСД, все возможные состояния каждой из вершин, а также вероятности пребывания вершины в каждом из состояний. Посмотреть эти вероятности можно, дважды щелкнув на имени вершины в списке вершин. Можно также просмотреть вероятности пребывания всех вершин во всех возможных состояниях, нажав пиктограмму списка вершин [expand node list tool], вторую слева на панели инструментов (рис. 12).

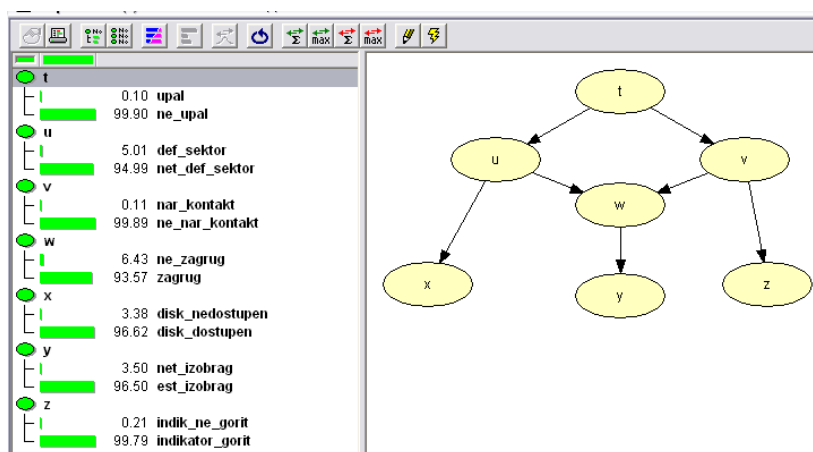


Рис. 12. Просмотр всех вероятностей

Теперь экспертная система, построенная на основе БСД, готова к эксплуатации. Например, возникла следующая проблема: на мониторе нет изображения и требуется найти причины, по которым это могло произойти. Т.е., в базу знаний экспертной системы добавляется новый факт о том, что нет изображения (вероятность пребывания вершины u в состоянии «нет изображения» - 100%). В результате этого происходит распространение вероятностей в БСД. Реализация запроса такого типа к экспертной системе на основе БСД выполняется следующим образом:

- Раскройте список всех вершин, нажав [expand node list tool].
- Введите факт отсутствия изображения, дважды щелкнув на состоянии «нет изображения» для вершины u (рис.13).

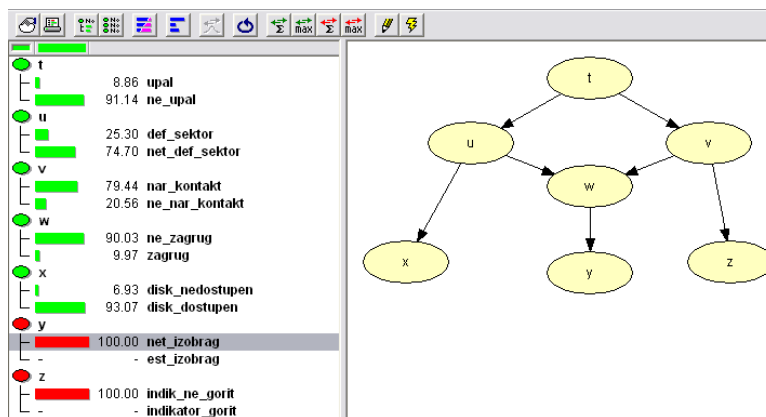


Рис.13. Введение нового факта в БСД

- Нажмите пиктограмму распространения (рис.14) на панели инструментов для распространения данного факта на всю БСД. Оцените полученные вероятности.



Рис.14. Пиктограмма распространения вероятностей

Допустим, на вход системы подается два факта: нет изображения и индикатор не горит. Результаты распространения вероятностей по байесовской сети доверия для этого случая приведены на рис.15.

Вероятность того, что причина данной ситуации то, что компьютер не загружается равна 90,03%, а то, что нарушен контакт – 79,44%.

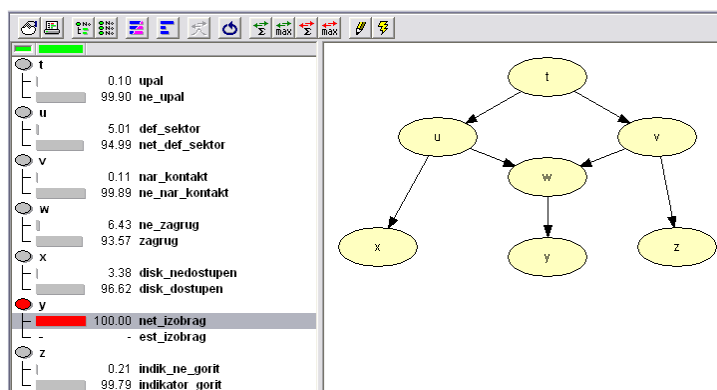


Рис.15. Результат распространения вероятностей

Ранее панель списка вершин использовалась для ввода фактов и нахождения вероятностей. То же самое можно сделать, используя окно контроля [monitor windows]. Окна контроля показывают ту же информацию, что и панель списка вершин, но появляется возможность располагать эти окна рядом с соответствующими узлами БСД в панели отображения сети. Можно открыть такое окно для любой вершины, но лучший вариант - это открывать лишь те окна, которые представляют особый интерес. Откроем окна контроля для некоторых вершин и повторим расчет. Для этого инициализируем БСД, т.е. приведем ее к исходному состоянию, нажав пиктограмму начальной инициализации в виде кольца со стрелкой [initialize tool button], расположенную слева от пиктограммы распространения. Далее необходимо выбрать вершины, для которых будут открываться окна. Если таких вершин несколько, то при их выборе необходимо удерживать клавишу SHIFT. Затем необходимо выбрать "Show Monitor Windows" (Показывать окна контроля) из пункта меню "View" (Вид) (рис.16, 17).

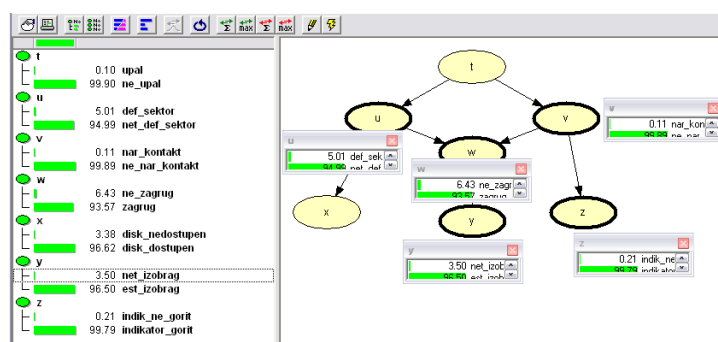


Рис. 16. Открытие окон контроля

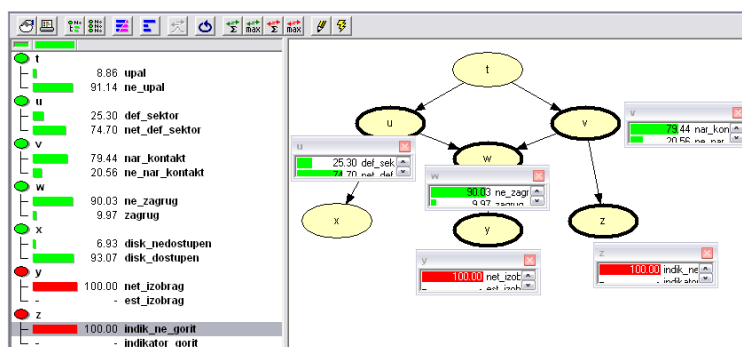


Рис. 17. Результат распространения вероятностей

Таким образом, экспертная система, спроектированная на основе БСД, на основании введенных фактов, описывающих конкретную ситуацию, может сделать вывод о возможных причинах, приведших к её возникновению, и определить вероятности, с которыми эти причины влияют на ситуацию.

Рассмотрим еще один тип вершин в БСД – непрерывные вершины.

Теория математической статистики различает два типа случайных величин: дискретные и непрерывные. Дискретная случайная величина - это изолированные значения исследуемого параметра, которые можно заранее пересчитать. Дискретные величины отделены друг от друга промежутками, где нет других возможных значений соответствующего параметра. Дискретная величина задается перечнем всех ее возможных значений с соответствующими вероятностями, что характеризует закон распределения дискретной величины. Непрерывная – это случайная величина, которая может принять любое значение из конечного или бесконечного интервала. Непрерывные величины не отделены друг от друга и заполняют некоторый интервал, причем одно значение случайной величины нельзя отделить от другого промежутком, не содержащим возможного значения этой же случайной величины. При этом уже нельзя говорить о вероятности отдельного состояния, так как при бесконечно большом их числе вес каждого будет равен нулю. Поэтому распределение вероятностей для непрерывной случайной величины определяется иначе, чем в дискретном случае и для их характеристики используются функции и плотности распределения вероятностей.

Байесовские сети, в которых могут присутствовать как дискретные переменные, так и непрерывные, называются гибридными байесовскими сетями.

К распространенным видам распределения случайной величины относят равномерное, нормальное (гауссовское) и показательное. В БСД непрерывные вершины подчинены нормальному закону распределения, так как он наиболее часто встречается на практике и является предельным законом, к которому приближается ряд других законов. Также нормальное распределение позволяет учесть воздействие большого количества факторов на значения анализируемой случайной величины

Непрерывная случайная величина имеет нормальное распределение, если ее дифференциальная функция распределения (плотность распределения вероятности) имеет следующий вид:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(\frac{-(x-m)^2}{2 \cdot \sigma^2}\right) \quad \text{для } -\infty < x < \infty \quad (1)$$

Нормальное распределение определяется параметрами m и σ , называемыми математическим ожиданием и среднеквадратическим отклонением и обычно обозначается $N(m, \sigma)$. Нормальное распределение имеет график распределения вероятности, представленный на рис.2. В случае нормального распределения выражение вида (2) позволяет определить вероятность того, что случайная величина попадает на заданный интервал вещественной оси (a, b) .

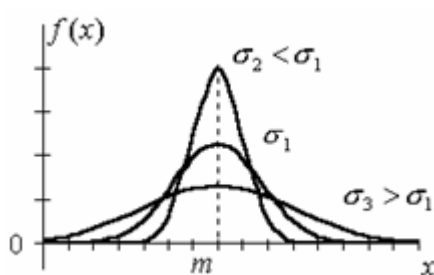


Рис.18. Графики плотности распределения вероятностей гауссовских переменных

Функция распределения и плотность распределения вероятностей являются наиболее полными характеристиками случайных величин. Однако во многих задачах практики оказывается трудно или даже невозможно полностью описать функцию распределения вероятностей.

В то же время для решения многих задач достаточно знать лишь некоторые параметры, характеризующие случайную величину с той или иной точки зрения. Наиболее распространёнными числовыми характеристиками (или моментами) случайных величин являются математическое ожидание и дисперсия, которые определяются следующими математическими выражениями:

Для дискретных случайных переменных, когда пространство исходов эксперимента Z является конечным множеством	Для непрерывных случайных переменных, когда пространство исходов эксперимента Z содержит бесконечное множество точек
$M(z) = m_z = \bar{z} = \sum_{z \in Z} z \cdot p(z)$	$M(z) = m_z = \bar{z} = \int_{-\infty}^{\infty} z \cdot f(z) \cdot dz$
$D(z) = \mu_2(z) = \sum_{z \in Z} (z - m_z)^2 \cdot p(z)$	$D(z) = \mu_2(z) = \int_{-\infty}^{\infty} (z - m_z)^2 \cdot f(z) \cdot dz$

Математическое ожидание является величиной, вокруг которой группируются значения случайной переменной. Дисперсия характеризует отклонение значений случайной величины от математического ожидания, то есть является характеристикой рассеивания случайной величины. Чем меньше дисперсия, тем более тесно группируются отдельные значения случайной величины вблизи математического ожидания.

Однако в ряде случаев дисперсия оказывается неудобной для практического использования, так как имеет размерность квадрата случайной величины. Поэтому в качестве характеристики рассеивания случайной величины часто используют корень квадратный из дисперсии, получивший название среднеквадратического отклонения:

$$\sigma = \sqrt{D} \quad (3)$$

В настоящее время существует ряд программных реализаций оболочек экспертных систем на основе БСД, которые позволяют оперировать не только дискретными, но и непрерывными случайными переменными. К числу таких программных средств относится и Hugin. Однако при использовании БСД, содержащих как непрерывные, так и дискретные переменные существует ряд ограничений: дискретные переменные не могут иметь непрерывных родителей; непрерывные переменные должны иметь нормальный закон распределения, условный на значениях родителей; распределение непрерывной переменной с дискретными и непрерывными родите-

лями является нормальным распределением с математическим ожиданием, зависящим от непрерывных и дискретных родителей и среднеквадратическим отклонением, не зависящим от непрерывных родителей. Это ограничение гарантирует возможность точного вывода.

Рассмотрим пример проектирования простейшей ЭС на основе БСД с непрерывными и дискретными событиями

Пусть требуется построить ЭС, позволяющую оценивать суммарные производственные затраты в зависимости от использования и загрузки трёх групп оборудования (например, трех пилорам). Такая ЭС поможет в выборе наиболее рациональной загрузки оборудования, в обоснованном решении об аренде необходимого оборудования и допустимых арендных платежах и многое другое, необходимое менеджеру или инженеру по деревообработке.

При анализе предметной ситуации эксперты установили, что в состав суммарных производственных затрат (без учёта зарплаты и начислений) входят:

- прямые производственные затраты на каждую группу оборудования за исследуемый календарный период, которые зависят как от количества используемых групп оборудования, так и от времени работы каждой из групп в течение исследуемого периода времени, т.е. от степени загрузки каждой из групп;
- расходы на амортизацию каждой из групп оборудования, зависящие как от её балансовой стоимости, так и установленных норм амортизации;
- арендная плата за участок при каждой из групп оборудования, используемый для складирования сырья и продукции, которая зависит как от площади участка, так и от ставок арендной платы.

Построение любой модели БСД начинается с выделения основных объектов и событий предметной области, анализа возможных состояний каждого из событий и установления причинно-следственных связей между ними. Так, в нашем примере, исходя из мнения экспертов, можно заключить, что на суммарные производственные затраты оказывают влияние:

- степень загрузки каждой из групп оборудования в течение исследуемого периода;
- значения нормы амортизационных отчислений;

- ставка арендной платы за участок, используемый определенной группой оборудования.

При этом модель БСД будет иметь вид, приведенный на рис.1, где одинарные овалы соответствуют дискретным событиям, а двойные овалы - непрерывным событиям (гауссовским переменным).



Рис.19 – Модель байесовской сети доверия

Для того, чтобы данная качественная модель превратилась в полную БСД, необходимо определить ее количественные характеристики. Для этого необходимо тщательно проанализировать каждое из событий. Так, вершина "Загрузка оборудования" соответствует дискретному событию, которое характеризуется тремя возможными состояниями. Вероятность пребывания в каждом из них определяется степенью загрузки каждой из групп оборудования, при условии, что суммарная загрузка всего оборудования равна единице. Если считать, что все группы оборудования загружены равномерно, то распределение вероятностей для этой вершины будет иметь вид, представленный в табл.1. При этом следует отметить, что возможны и любые другие исходные распределения вероятностей, учитывающие различные варианты загрузки оборудования.

Таблица 1 – Таблица безусловных вероятностей

Р(«Загрузка оборудования» «Пилорама 1»)	=	Р(«Загрузка оборудования» «Пилорама 2»)	=	Р(«Загрузка оборудования» «Пилорама 3»)
0.333		0.333		0.333

Полученные от экспертов знания о том, что ставка аренды 1 га земли в среднем составляет 2500 у.е. и колеблется в пределах $\pm 10\%$, т.е. принимает значения 2500 ± 250 у.е., а норма амортизации может находиться в пределах 5 - 10 % от балансовой стоимости, т.е. принимать значения 7,5 $\pm$ 2,5% (или 0,075 $\pm$ 0,025 относительных единиц), позволяют определить параметры ещё двух вершин - "Ставка арендной платы" и "Норма амортизации". Предполагая, что эти вершины шансов являются непрерывными случайными переменными с нормальным законом распределения, необходимо задать параметры этого распределения для каждой из вершин, которые представлены в табл. 2.

Таблица 2 – Параметры распределения непрерывных случайных величин

	Вершина «Ставка арендной платы»	Вершина «Норма амортизации»
математическое ожидание(m)	2500	0,075
дисперсия ($D = s^2$)	62500 (250 ²)	0,000625 (0,025 ²)

Что касается вершины «Производственные затраты», то она характеризуется случайной переменной, условно нормальной на значениях родителей (т.е. на значениях трёх других вершин примера). Следует отметить, что в общем случае распределение вероятностей для вершин, аналогичных вершине «Производственные затраты» является не просто нормальным, а смешанным нормальным распределением. Т.е., представляет собой весовую сумму распределений, для каждого из которых должен быть задан список его параметров:

- математические ожидания и дисперсии для распределений, описывающих степень влияния дискретных родителей;
- весовые коэффициенты, учитывающие степень влияния на математическое ожидание непрерывных родителей.

Если теперь, используя экспертное оценивание, предположить, что балансовая стоимость каждой из пилорам составляет 50000, 40000 и 30000 у.е., площадь арендуемых участков, закрепляемая за ними равна 0,6; 0,5 и 0,4 га, оценка прямых затрат на поддержание нормальной работы каждой из пилорам в среднем составляет 3000, 3200 и 3500 у.е. и получена с 5% точностью, то степень влияния родительских вершин на «Производственные затраты» можно представить таблицей 3.

Таблица 3 - Параметры, определяющие распределение вероятностей для вершины «Производственные затраты»

«Загрузка оборудования»	Пилорама 1	Пилорама 2	Пилорама 3
математическое ожидание (m)	3000	3200	3500
дисперсия ($D = s^2$)	22500 (1502)	25600 (1602)	30625 (1752)
«Норма амортизации»	50000	40000	30000
«Ставка аренды»	0,6	0,5	0,4

На основе проведенного экспертного оценивания предметной области можно теперь переходить к реализации БСД в системе Hugin и построения на ее базе ЭС, которая позволит получить оценку любых ее состояний, при любых произвольно задаваемых свидетельствах.

Реализация модели БСД с непрерывными и дискретными состояниями включает в себя несколько этапов, первым из которых является этап построения вершин шансов и установления связей между ними. Для того чтобы включить в состав БСД новые дискретные или непрерывные вершины шансов следует на панели инструментов выбрать кнопку с пиктограммой в виде одинарного (Discrete Chance Tool)

или двойного эллипса (Continuous Chance Tool), соответственно (рис.20), затем подвести указатель мыши к тому месту на панели отображения сети, где будет располагаться вершина, и нажать левую клавишу мыши.



а)



б)

Рисунок 20 – Выбор типа вершин БСД (а – дискретные вершины, б – непрерывные)

После того, как все необходимые вершины будут включены в модель БСД, необходимо определить причинно-следственные связи между событиями, соответствующими вершинам БСД. Для этого надо выбрать на панели инструментов пиктограмму со стрелочкой (Link Tool), подвести курсор к вершине С1 и, удерживая кнопку мыши нажатой, провести линию к С3. Затем проделать то же с другими вершинами (рис.21).

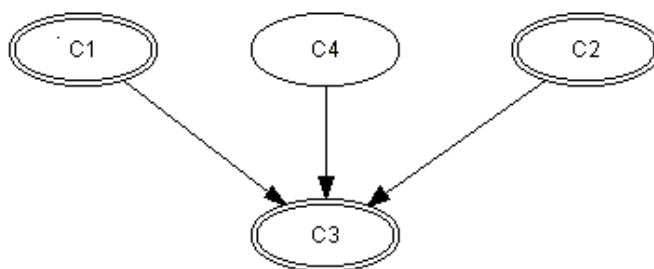


Рисунок 21 – БСД с причинно-следственными связями

В тех случаях, когда необходимо добавить несколько однотипных элементов БСД, можно на панели инструментов выбрать нужный компонент, затем переместить курсор на рабочую область, нажать клавишу Shift и, удерживая ее нажатой, построить необходимое количество элементов.

Вторым этапом процесса реализации модели БСД в системе Nugin является присвоение вершинам шансов имен и ярлыков. С этой целью необходимо выполнить следующую последовательность действий:

- выделить курсором мыши нужную вершину;
- выбрать на панели инструментов пиктограмму «Node Properties» или, нажав правую кнопку мыши, выбрать эту команду в контекстном меню;
- в форме свойств (рис.22) в полях Name и Label задать новое имя и метку вершины (наименования только латинскими буквами без пробелов, метки - в зависимости от подгруженных шрифтов, но желательно латинскими).

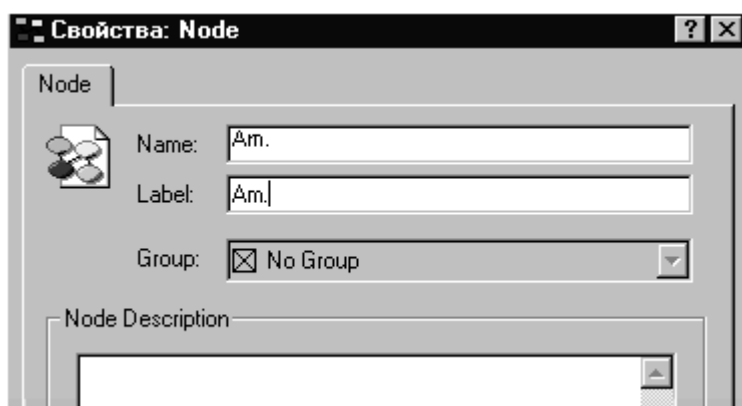


Рисунок 22 – Присвоение вершинам имен и ярлыков

Если заданная метка вершины слишком длинная и не помещается в стандартные границы овалов вершин шансов, то имеется возможность изменить их параметры. Для этого в пункте «Network» главного меню необходимо выбрать опцию «Network Properties» (или нажать Alt-Enter) и затем в свойствах «Display» окна «Node Size» задать новые размеры узлов сети.

Окончательный вариант модели БСД с непрерывными и дискретными вершинами шансов для рассматриваемой задачи будет иметь вид, приведенный на рис.23, где Am.- амортизация; Ar.- арендная плата; Z_O - загрузка оборудования; P_Z - производственные затраты.

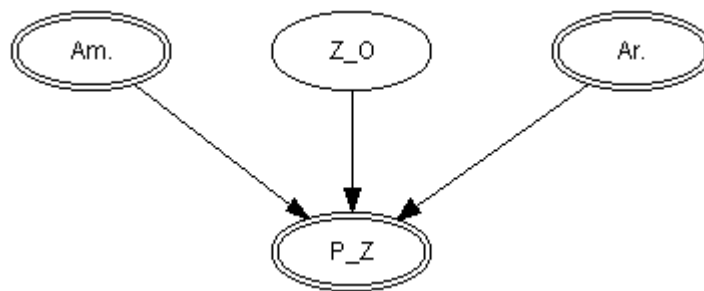


Рисунок 23 – Окончательный вариант БСД

На третьем этапе реализации БСД необходимо дать количественное описание сети и всех ее элементов, т.е. определить все возможные состояния каждой из вершин сети и задать их вероятностные характеристики.

Для задания количественных параметров дискретной вершины шансов необходимо выполнить следующую последовательность действий:

- двойным щелчком выделить вершину, соответствующую загрузке оборудования (Z_O), в результате чего на панели редактирования появится таблица состояний (State) дискретной вершины (рис.24);
- кнопкой «Add State» необходимо добавить нужное число состояний;
- для переименования состояния необходимо выделить двойным щелчком левой клавиши мыши редактируемую область и ввести новое название состояния;
- проставить соответствующие априорные вероятности возможных состояний загрузки оборудования (рис.24).

State 0	1
---------	---

Pilorama 1	0.3333
Pilorama 2	0.3333
Pilorama 3	0.3333

Рисунок 24 – Задание количественных параметров дискретной вершины

Для задания количественных параметров непрерывных вершин шансов необходимо выполнить следующую последовательность действий:

- двойным щелчком выделить вершину, соответствующую ставке арендной платы за 1 га (Ar);
- задать значения математического ожидания (Mean) и дисперсии (Variance) нормального закона распределения вероятностей для непрерывного случайного события, соответствующего значению арендной ставки (рис.25);
- проделать аналогичную операцию с вершиной (Am), определяющей случайное событие, соответствующее значению нормы амортизационных отчислений (рис.9);
- двойным щелчком левой клавиши мыши выбрать вершину шансов, соответствующую суммарным производственным затратам (P_Z);
- используя данные табл.2, заполнить таблицу условных вероятностей для вершины производственных затрат (рис.25).

Mean	2500
Variance	62500

Рисунок 25 – Задание параметров для вершины «Арендная ставка»

Mean	0.075
Variance	0.000625

Рисунок 26 - Задание параметров для вершины «Норма амортизации»

Z_O	Pilorama 1	Pilorama 2	Pilorama 3
Mean	3000	3200	3500
Variance	22500	25600	30625
Am.	50000	40000	30000
Ar.	0.6	0.5	0.4

Рисунок 27 - Таблица условных вероятностей для вершины «Производственные затраты»

На этом формирование БСД для рассматриваемого примера заканчивается и можно переходить к проверке, исследованию и эксплуатации спроектированной сети в качестве экспертной системы.

Для тестирования и исследования сформированной модели необходимо ее откомпилировать и запустить на выполнение, т.е. из режима редактирования перейти в режим вычислений. Для этого необходимо нажать кнопку «Run» на панели инструментов. В результате выполнения этой команды на экране появится модель БСД, представленная в двух видах: в виде списка вершин («Node List») и в виде графического отображения сети.

Когда все состояния случайных событий описаны и модель переведена в режим вычислений, можно исследовать влияние одних компонентов модели на другие. При этом надо владеть технологией работы со списком вершин и графическом отображении их состояний на модели БСД.

При переходе в режим вычислений, в левой части экрана отображается список всех вершин исследуемой БСД, который будет иметь вид, аналогичный приведенному на рис.28.

Для того чтобы вызвать на экран состояния той или иной вершины (рис.29) надо щелкнуть левой клавишей мыши по ее изображению в списке вершин или дважды щелкнуть левой клавишей мыши на овале вершины в панели графического отображения сети.

Повтор этой операции вернет список в исходное состояние (рис.28). Эту процедуру можно проделывать и с отдельными (выборочными) вершинами, и со всеми одновременно.

Для одновременного вывода на экран всех состояний всех вершин модели следует воспользоваться кнопкой «Expand Node List» (раскрыть список вершин) на панели инструментов. На экране появится результат, аналогичный рис.30.

Для сворачивания раскрытого списка используйте кнопку «Collapse Node List». При этом будут свернуты все вершины модели независимо от того, использовался ли инструмент «Expand Node List» или вручную раскрывались выбранные вершины.

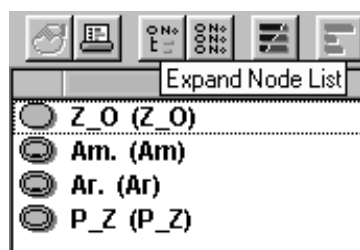


Рисунок 28 - Список всех вершин исследуемой БСД

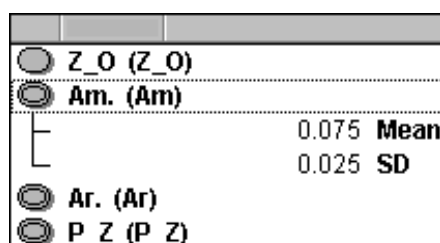


Рисунок 29 – Раскрытие произвольной вершины

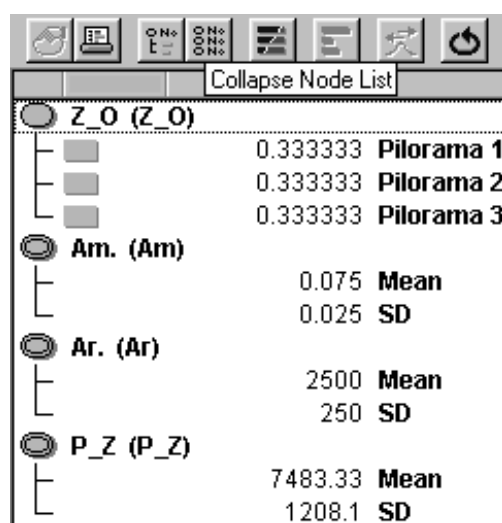


Рисунок 30 - Одновременный вывод на экран всех состояний всех вершин

С правой частью окна БСД, где представлено графическое отображение сети можно работать так же как и со списком вершин, просматривая как числовые параметры, так и графики плотностей распределения вероятностей каждой из вершин

БСД. Рассмотрим технологию работы с графическим отображением параметров вершин:

- необходимо выбрать вершину, состояние которой требуется отобразить на экране, и выделить ее щелчком левой клавиши мыши;
- нажатием правой клавиши мыши вызывается на экран контекстное меню, в котором необходимо выбрать опцию «Show Monitor», в результате чего на экране появится графическое окно с состояниями вершины;
- необходимо нажать кнопку с изображением графика (это возможно только для непрерывной случайной величины);
- для обновления графика плотности распределения вероятностей непрерывной случайной величины необходимо нажать кнопку «Update Monitor Graphs» на панели инструментов или в нижнем правом углу графического окна вершины;
- для того чтобы изменить интервал построения графика можно установить нужный диапазон значений среднеквадратических отклонений, отображаемых влево и вправо от математического ожидания. Эти значения выбираются из двух выпадающих списков окна графического отображения вершины (рис.31);
- после любого изменения параметров графической области необходимо нажимать кнопку Update Monitor Graphs.

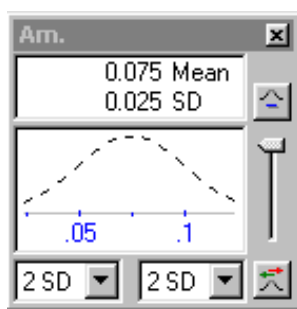


Рисунок 31 – Окно графического отображения вершины

Кроме этого система Hugin позволяет одновременно для всех вершин сети отобразить их параметры в виде графических окон (рис.15). Для этого необходимо:

- выбрать все вершины на панели отображения сети, что возможно сделать либо мышью при нажатой клавише Shift, либо вызовом на экран контекстного меню (нажав правую клавишу мыши) и выбором в нем опции «Select All»;

- вызвать на экран контекстное меню и выбрать в нем опцию «Show Monitor»;
- нажать на пиктограмму с символом добавления графика и затем, когда появится окно графика, нажать кнопку «Update Monitor Graphs» на панели инструментов (в случае работы со всеми вершинами сети этот способ более рационален);
- если графики недостаточно наглядные, изменить интервалы отклонений от математических ожиданий и нажать кнопку «Update Monitor Graphs».

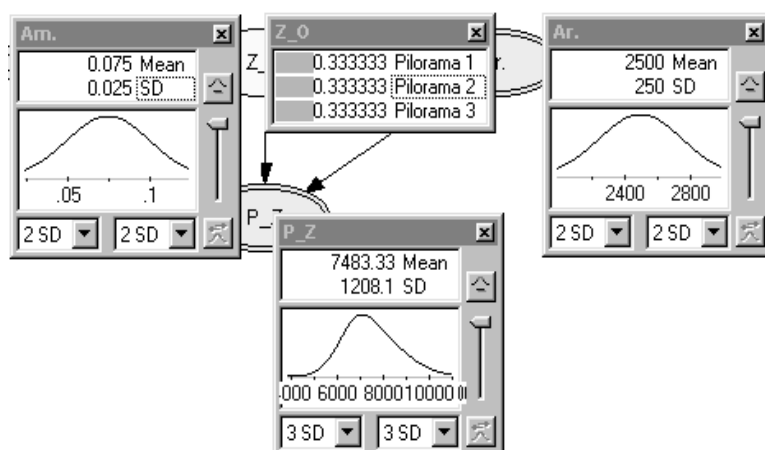


Рисунок 32 – Отображение параметров для всех вершин БСД

Процесс эксплуатации БСД как экспертной системы заключается в автоматическом распространении вероятностей (логическом выводе), который осуществляется в системе при поступлении в нее свидетельств о наступлении тех или иных событий или вероятностях их возникновения. Для того чтобы задать в системе какое-либо свидетельство о событии, необходимо:

- выбрать требуемое событие в списке вершин или в соответствующем окне графического отображения вершины;
- если это дискретное событие и свидетельство соответствует наступлению какого-либо одного из его состояний, то необходимо двойным щелчком левой клавиши мыши зафиксировать одно из его состояний (вероятность его наступления в этом случае окажется равна 1);
- если это дискретное событие и свидетельство соответствует возможности наступления нескольких его состояний с заданной вероятностью, то необходимо нажать кнопку «Enter Likelihood» на панели инструментов (рис.32) и в появившемся

на экране окне «Likelihood» задать распределение вероятностей для всех состояний задаваемого свидетельства;

- если это непрерывное событие, то двойной щелчок левой клавиши мыши на его графическом изображении вызовет на экран окно «Continuous Finding» (рис.33), где необходимо в строке Evidence/Value задать новое числовое значение математического ожидания этого события, соответствующее задаваемому свидетельству;

- после задания одного или группы свидетельств, следует провести распространение сумм вероятностей на все составляющие БСД, нажав кнопку «Prop Sum Normal» панели инструментов;

- для просмотра изменений на графиках необходимо использовать кнопку «Update Monitor Graphs»;

- для приведения системы в исходное состояние (т.е. отмены введенных свидетельств), необходимо использовать кнопку «Initialize» на панели инструментов.



Рисунок 33 – Задание дискретных событий

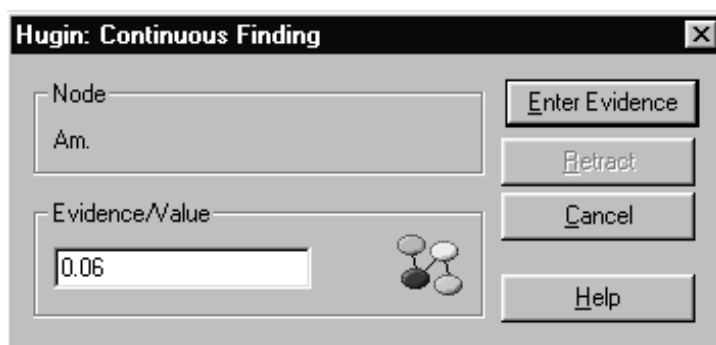


Рисунок 34 – Задание непрерывных событий

Допустим, требуется оценить возможное изменение суммарных производственных затрат при загрузке только одной из групп оборудования при допустимом небольшом снижении норм амортизации. Для ответа на такого типа запрос, необходимо в исходную БСД ввести два новых свидетельства:

- свидетельство о том, что норма амортизации оборудования в рассматриваемом случае может составлять 6% (т.е. математическое ожидание данного непрерывного свидетельства будет равна 0.06, а дисперсия равна 0);

- свидетельство об использовании только одной, а именно 1-ой группы оборудования (т.е. $P(\text{Pilorama1})=1$, $P(\text{Pilorama2})=0$, $P(\text{Pilorama3})=0$).

Проведя распространение распределения вероятностей и параметров гауссовских законов распределения на остальные составляющие БСД, можно увидеть, что производственные затраты увеличились по сравнению с первоначальным результатом (рис.35).

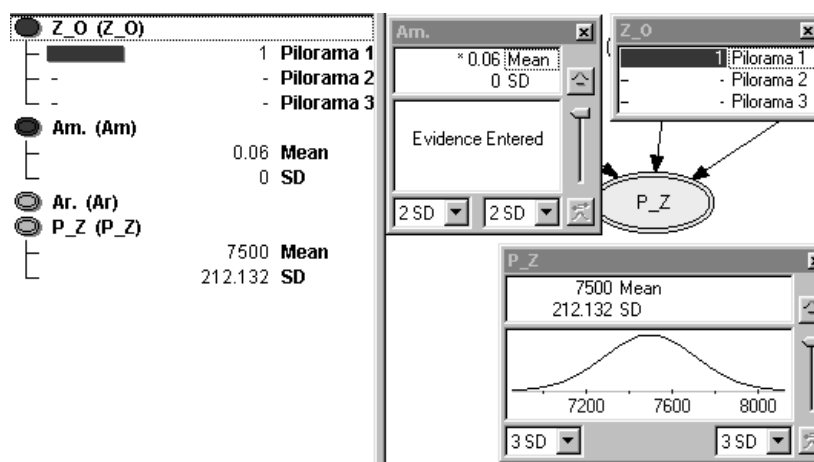


Рисунок 35 – Результат работы системы

Проанализировав подобным образом использование других пилорам, можно сделать вывод о том, что при заданном значении нормы амортизации и известных параметрах другой непрерывной случайной величины - арендной платы, наиболее эффективным является использование третьей пилорамы.

Однако можно задавать свидетельства не только вершинам, влияющим на конечный результат, но и самому конечному результату. Это позволит исследовать обратный процесс и узнать, какие состояния должны иметь исходные вершины, чтобы получился заданный результат.

Задание на лабораторную работу

1. Изучив теоретический материал и основные приемы работы в системе Hugin, самостоятельно реализовать в этой среде экспертную систему в соответствии с примером, разобранным в методичке.

2. Определить предметную область, для которой Вы хотели бы разработать ЭС. Описать проблемную область в соответствии с предложенным в теоретическом введении подходом. Спроектировать байесовскую сеть доверия и описать вершины и дуги этой сети, задав соответствующие вероятности. С помощью Hugin создать свою экспертную систему, после чего протестировать полученную ЭС в режиме консультации.

Лабораторная работа №3

Оболочки экспертных систем как средство проектирования. Разработка экспертной системы на основе байесовской системы логического вывода

Целью работы является изучение основ проектирования экспертных систем на базе байесовской системы логического вывода (субъективной вероятности).

Плохо поставленные (некорректные) задачи существуют во всех предметных областях, где могут быть эффективны интеллектуальные системы как средство построения информационных систем нового поколения. В области некорректных задач точные знания нельзя получить, либо их нельзя поучить сразу, поэтому, как правило, необходим подход, суть которого в постепенном приближении к полному набору знаний. Другими словами, знания чаще всего нечетки. Для того чтобы интеллектуальные системы вышли за рамки простых символьных выводов и приблизились к мышлению человека, необходимы методы представления нечетких знаний и механизм выводов, работающий в их среде.

В настоящее время модели представления знаний являются предметом исследований и разработок в среде узких специалистов, в основном программистов и математиков, в то время как потребность в таких моделях ощущается практически во

всех предметных областях. Это объясняется необходимостью построения экспертных систем, структура которых неразрывно связана с формами представления знаний, определяемых в свою очередь особенностями данной предметной области.

Основным назначением ЭС является разработка программных средств, которые при решении задач, трудных для человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым человеком-экспертом. ЭС используются для решения так называемых неформализованных задач, общим для которых является то, что:

- задачи не могут быть заданы в числовой форме;
- цели нельзя выразить в терминах точно определённой целевой функции;
- не существует алгоритмического решения задачи;
- если алгоритмическое решение есть, то его нельзя использовать из-за ограниченности ресурсов (время, память).

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

Экспертная система – это программное средство, использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. Основу ЭС составляет база знаний (БЗ) о предметной области, которая накапливается в процессе построения и эксплуатации ЭС. Накопление и организация знаний – важнейшее свойство всех ЭС.

Данные и знания, которыми оперирует ЭС, редко бывают абсолютно точными и достоверными. Присущая знаниям неопределенность может иметь разнообразный характер, и для ее описания используется широкий спектр формализмов. Рассмотрим один из типов неопределенности в данных и знаниях – их неточность. Будем называть высказывание *неточным*, если его истинность (или ложность) не может быть установлена с определенностью. Основопологающим понятием при построении моделей *неточного вывода* является понятие вероятности, поэтому все описываемые далее методы связаны с вероятностной концепцией.

Модель оперирования с неточными данными и знаниями включает две составляющие: язык представления неточности и механизм вывода на неточных знаниях.

Для построения языка необходимо выбрать форму представления неточности (например, скаляр, интервал, распределение, лингвистическое выражение, множество) и предусмотреть возможность приписывания меры неточности всем высказываниям.

Механизмы оперирования с неточными высказываниями можно разделить на два типа. К первому относятся механизмы, носящие «присоединенный» характер: пересчет мер неточности как бы сопровождает процесс вывода, ведущийся на точных высказываниях. Для разработки присоединенной модели неточного вывода в основанной на правилах вывода системе необходимо задать функции пересчета, позволяющие вычислять:

а) меру неточности антецедента правила (его левой части) по мерам неточности составляющих его высказываний;

б) меру неточности консеквента правила (его правой части) по мерам неточности правила и посылки правила;

в) объединенную меру неточности высказывания по мерам, полученным из правил.

Введение меры неточности позволяет использовать в процессе вывода возможность объединения силы нескольких свидетельств, подтверждающих или опровергающих одну и ту же гипотезу. Для объединения свидетельств требуется функция пересчета, занимающая центральное место в пересчете.

Для механизмов оперирования с неточными высказываниями второго типа характерно наличие схем вывода, специально ориентированных на используемый язык представления неточности. Как правило, каждому шагу вывода соответствует пересчет мер неточности, обусловленный соотношением на множестве высказываний (соотношением может быть элементарная логическая связь, безотносительно к тому, является ли это отношение фрагментом какого-либо правила). Таким образом, механизмы второго типа применимы не только к знаниям, выраженным в форме правил. Вместе с тем для них, как и для механизмов «присоединенного» типа, одной из главных является проблема объединения свидетельств.

Рассмотрим наиболее важные термины, которые необходимо знать при разработке ЭС на основе байесовской системы логического вывода.

Пусть H – событие, заключающееся в том, что данная гипотеза верна. Пусть E – событие, заключающееся в том, что наступило определенное доказательство (свидетельство), которое может или не может подтвердить правильность указанной гипотезы.

$P(H)$ – априорная вероятность истинности гипотезы H , т.е. вероятность наступления H без учета факта существования E .

$P(E)$ – вероятность того, что событие E произошло.

Вероятности являются числами в диапазоне от 0 до 1. Если вероятность равна 0, то данное событие не произойдет. Если вероятность равна 1, то данное событие происходит всегда.

$P(\text{не } H) = 1 - P(H)$ – вероятность того, что событие H ложно.

$P(H:E)$ или $P(H:\text{не}E)$ – апостериорная вероятность гипотезы H , т.е. вероятность H при условии, что известен факт существования E .

Два события независимы, если вероятность их совместного наступления равна произведению вероятностей наступления каждого события в отдельности. $P(E_1 \& E_2)$ – совместная вероятность наступления как события E_1 так и события E_2 . Т.е. события E_1 и E_2 независимы тогда и только тогда, когда

$$P(E_1 \& E_2) = P(E_1) \cdot P(E_2). \quad (1)$$

$P(H:E)$ – условная вероятность наступления события H , определяемая с учетом того, что событие E уже наступило. Если события H и E независимы, то $P(H:E) = P(H)$. В общем случае:

$$P(H:E) = P(H \& E) / P(E). \quad (2)$$

Аналогично,

$$P(E:H) = P(E \& H) / P(H). \quad (3)$$

Поэтому:

$$P(H:E) = P(E:H) \cdot P(H) / P(E). \quad (4)$$

Тогда теорему Байеса можно записать:

$$P(H:E)=P(E:H) \cdot P(H) / (P(E:H) \cdot P(H)+P(E:\text{не}H) \cdot P(\text{не}H)). \quad (5)$$

Для двух или более событий E_1 и E_2 при условии их независимости:

$$P(E_1 \& E_2:H)= P(E_1:H) \cdot P(E_2:H) \quad (6)$$

Примем, что для некоторого события H существует большое число отдельных свидетельств, подтверждающих или не подтверждающих его, которые последовательно выявляются в процессе функционирования ЭС – $E_1, E_2, \dots, E_n$. Если бы все они были выявлены одновременно и не зависели друг от друга, то можно было бы вычислить:

$$P(E:H)= P(E_1:H) \cdot P(E_2:H) \cdot \dots \cdot P(E_n:H), \quad (7)$$

а затем вычислить $P(H:E)$, где E - событие, заключающееся в том, что «все E_i наступили». Аналогично можно было бы определить $P(\text{не}E:H)$ как произведение всех $P(\text{не}E_i:H)$.

Но можно действовать поэтапно, суммируя отдельные свидетельства и их влияние на условную вероятность по мере наступления отдельных E_i . Это можно сделать, используя априорные и апостериорные вероятности следующим образом:

- 1) $P(H)$ – априорная вероятность события H .
- 2) Для данного свидетельства E_i запишем $P(E_i:H)$. и $P(E_i:\text{не}H)$.
- 3) Используя теорему Байеса, подсчитаем $P(H:E_i)$ или $P(H:\text{не}E_i)$ в зависимости от исхода E_i , т.е. вычислим апостериорную вероятность события H .
- 4) Переобозначим текущую апостериорную вероятность события H как новую априорную вероятность H .
- 5) Выберем новое свидетельство E_i для рассмотрения и перейдем к шагу 1.

С учетом байесовской системы логического вывода примем, что большая часть информации (гипотезы и свидетельства) не является абсолютно точной, а носит вероятностный характер, и в соответствии с этим определим способ её представления и хранения.

Таблица 1 - Формат хранения свидетельств

Номер	Свидетельства
1	Свидетельство_1
2	Свидетельство_2
....	
N	Свидетельство_N

Этот формат данных будет использоваться для хранения свидетельств. Суть в том, что компьютер задает множество вопросов, содержащихся в виде символьных строк <Свидетельство_1>, <Свидетельство_2> и т.д. Например, Свидетельство_1 может означать строку "Много ли вы кашляете?" или "Установлен ли драйвер?" в зависимости от предметной области, для которой разрабатывается ЭС.

Таблица 2 - Способ хранения информации о гипотезах

Номер	Гипотеза	P(H)	[j, P(E:H), (E:неH)]
1	Гипотеза_1	P(H1)	[j, P(E:H ₁), (E:неH ₁)]
2	Гипотеза_2	P(H2)	[j, P(E:H ₂), (E:неH ₂)]
...			
N	Гипотеза_N	P(Hn)	[j, P(E:H _n), (E:неH _n)]

Этот формат данных будет использоваться для хранения гипотез, т.е. возможного исхода и всей информации, относящейся к нему. Поле "Гипотеза" характеризует название возможного исхода. Следующее поле - P(H) - это априорная вероятность такого исхода, т.е. вероятность исхода в случае отсутствия дополнительной информации. После этого идет ряд повторяющихся полей из трех элементов. Первый элемент j — это номер соответствующего свидетельства. Следующие два элемента — P(E:H) и P(E:не H) — соответственно вероятности получения ответа "Да" на этот вопрос, если возможный исход верен и неверен.

В качестве примера рассмотрим фрагмент базы знаний ЭС в области медицины. Одна из возможных гипотез – грипп. Информация об этой гипотезе может храниться в следующем виде:

Таблица 3- Пример описания болезни

2010	Грипп	0.01	(1, 0.9, 0.01); (2, 1, 0.01); (3, 0, 0.01)
------	-------	------	--

В этой таблице содержится следующая информация: существует априорная вероятность $P(H)=0.01$, что любой наугад взятый человек болеет гриппом.

Допустим, программа задает вопрос 1 (симптом 1). Тогда имеем: $P(E:H)=0.9$ и $P(E:\text{не } H)=0.01$. Это означает, что если у пациента грипп, то он с вероятностью 0.9 ответит "да" на этот вопрос, а если у него нет гриппа, он ответит "да" с вероятностью 0.1. Очевидно, ответ "да" подтверждает гипотезу о том, что у него грипп. Ответ "нет" позволяет предположить, что человек гриппом не болеет.

Так же и во второй группе симптомов (2, 1, 0.01). В этом случае $P(E:H)=0.9$, т.е. если у человека грипп, то этот симптом должен присутствовать. Соответствующий симптом может иметь место и при отсутствии гриппа $P(E:\text{не } H)=0.01$, но это маловероятно.

Свидетельство 3 исключает грипп при ответе "да", потому что $P(E:H)=0$. Это может быть, например, вопрос: "Наблюдаете ли вы такой симптом на протяжении большей части жизни?"

Описывая алгоритм, можно разделить программу на несколько частей:

- 1) Ввод данных (свидетельства и гипотезы);
- 2) Просмотр данных на предмет нахождения априорной вероятности $P(H)$. Это делается для того, чтобы определить, какие вопросы (свидетельства) являются самыми важными, и выяснить, о чем спрашивать в первую очередь.
- 3) Программа находит самый важный вопрос и задает его. Систему можно организовать таким образом, что ответы будут вводиться либо в форме "да" или "нет", либо "да", "нет", "не знаю", либо с использованием шкалы (например, от -5 до +5), чтобы выразить степень уверенности в ответе.

4) Априорные вероятности заменяются новыми значениями при получении новых подтверждающих свидетельств;

5) Подсчитываются новые значения правил. Определяются также минимальное и максимальное значения для каждой болезни, основанные на существующих в данный момент априорных вероятностях и предположениях, что оставшиеся свидетельства будут говорить в пользу гипотезы или противоречить ей. Это делается для того, чтобы определить, стоит ли продолжать рассматривать данную гипотезу или нет. Гипотезы, которые не имеют смысла, отбрасываются. Те же из них, чьи минимальные значения выше определенного уровня, могут считаться возможными исходами. После этого - возврат к части 3.

Итак, в экспертных системах вероятности, требуемые для решения некоторой проблемы, обеспечиваются экспертами и запоминаются в базе знаний. Эти вероятности включают:

- априорные вероятности всех возможных гипотез $P(H)$;
- условные вероятности возникновения свидетельств при условии существования каждой из гипотез $P(E:H)$.

Для того, чтобы установить обоснованные значения для этих вероятностей, необходимо привлечение специалистов-экспертов в соответствующей проблемной области. Несмотря на то, что при разработке баз знаний ЭС, основанных на субъективной вероятности, предполагают условную независимость свидетельств для уменьшения числа требуемых вероятностных оценок, все же число оценок, требуемых для них, остаётся достаточно большим. Например, если некоторая область медицинских диагнозов имеет 100 диагнозов и 700 симптомов, то, по крайней мере, 70100 значений вероятностей (70000 условных + 100 априорных) должны быть получены. Поэтому разработка базы знаний - это очень трудоемкая и кропотливая работа и от её результата зависит адекватность работы ЭС.

Существенно упростить процесс построения ЭС можно при наличии оболочки экспертной системы - программы, позволяющей пользователю заполнить базу знаний определенной структуры конкретными знаниями проблемной области. В рамках данной лабораторной работы для реализации ЭС предлагается воспользоваться

программой «Малая ЭС», которую можно рассматривать как оболочку ЭС и которая использует байесовскую систему логического вывода. Эта программа предоставляет возможность создания собственной базы знаний и её последующего применения для проведения консультации с пользователем в какой-либо прикладной области (на которую настроена загруженная база знаний) с целью определения вероятностей возможных исходов и использует для этого оценку правдоподобности некоторых предпосылок, получаемую от пользователя.

Для создания новой или редактирования существующей базы знаний используется *Редактор базы знаний*, рабочее окно которого представлено на рис. 1.

База знаний представляет собой текстовый файл, содержащий три секции. В первой секции может содержаться описание базы знаний, имя автора, комментарии и др. (не более 10000 символов). Во второй секции содержатся свидетельства (каждое из которых заканчивается переносом строки). В третьей секции содержится информация о гипотезах в форме, описанной в табл. 2.

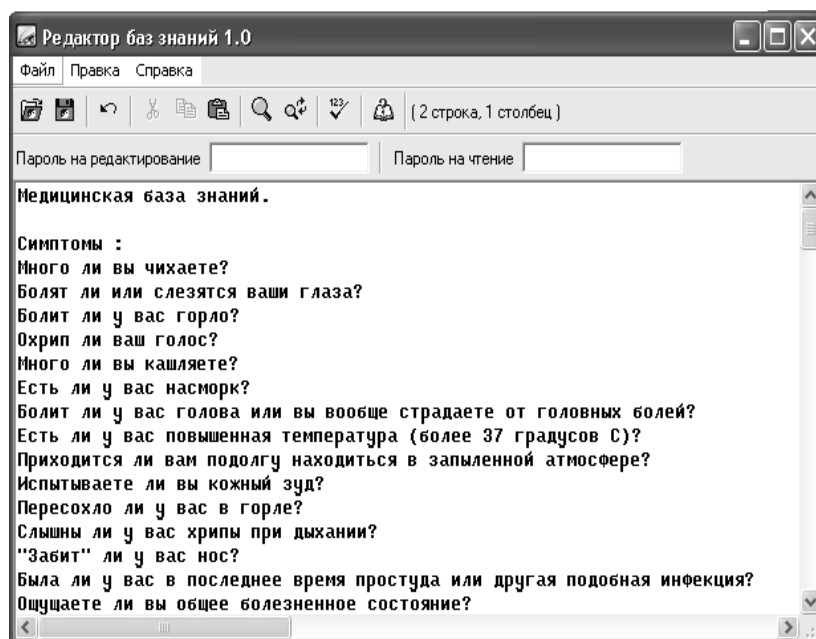


Рис.1. Рабочее окно *Редактора базы знаний*

После создания базы знаний можно приступать к работе с ЭС в режиме консультаций. Для этого необходимо запустить программу «Малая ЭС», загрузить из файла базу знаний (рис. 2), после чего нажать кнопку «Начало консультации». В

правой части окна появляется первый запрос системы. Существует два варианта ответа пользователя:

- можно задать коэффициент уверенности по шкале от -5 (точно нет) до +5 (точно да);
- можно ввести вероятность истинности свидетельства (от 0 до 1).

Переключение между двумя вариантами осуществляется кнопкой «Переключить способ ввода ответа». В окне «Настройка предпочтений» можно выбрать желаемый диапазон изменения коэффициента уверенности.

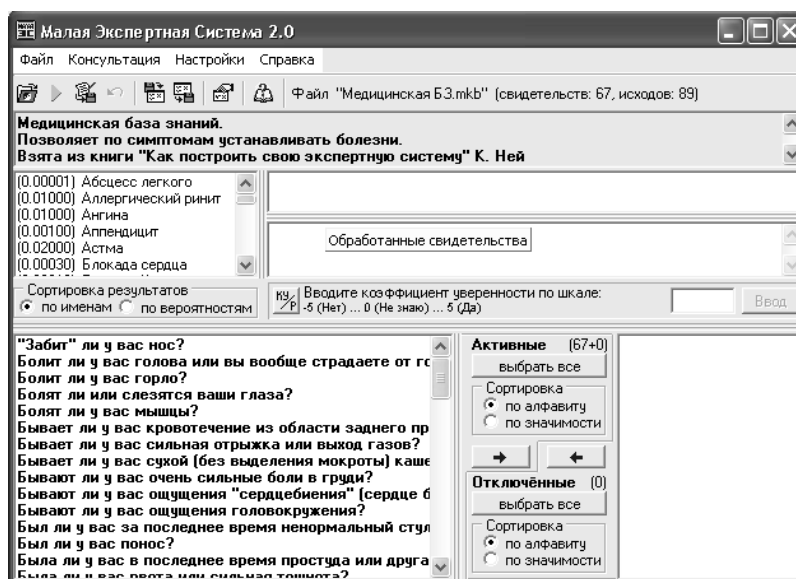


Рис.2. Рабочее окно оболочки «Малая ЭС»

Целью консультации является определение вероятностей всех возможных исходов. Список исходов с указанием текущих вероятностей отображается в левой части верхней половины окна программы. Для более удобного представления результатов их можно упорядочить либо по названию исходов (в алфавитном порядке), либо по значениям текущих вероятностей. Вероятности исходов могут быть выражены в процентах. Переключение формата вывода результатов осуществляется в окне «Настройка предпочтений». Здесь же можно сделать выбор относительно классификации результатов на достоверные и недостоверные (это влияет на показ результатов в окне: достоверные выделяются темно-красным цветом, недостоверные – серым, остальные – черным).

После того, как будет обработано последнее свидетельство, система подведет итог (выдаст количество обработанных свидетельств), а вероятности исходов в списке результатов примут окончательное значение. Теперь можно сделать вывод о возможности наступления интересующего исхода, или определить наиболее вероятный исход.

Задание на лабораторную работу

Определить предметную область, для которой Вы хотели бы разработать ЭС. Описать свидетельства и гипотезы в соответствии с предложенным в теоретическом введении подходом (в формате табл.1 и табл.2). С помощью «Редактора базы знаний» создать новую базу знаний, после чего протестировать полученную ЭС в режиме консультации.

Лабораторная работа №4

Решение задач распознавания буквенных образов с использованием программ нейросетевого моделирования

Цель работы: изучение основных принципов функционирования искусственных нейронных сетей на примере решения задачи распознавания буквенных образов, получение практических навыков по формированию обучающей выборки, обучению и тестированию нейронных сетей с использованием прикладных программ нейросетевого моделирования

Теория распознавания образов — раздел кибернетики, развивающий теоретические основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков.

Создание искусственных систем *распознавания образов* является сложной теоретической и технической проблемой. Традиционно задачи распознавания образов включают в круг задач искусственного интеллекта

Можно выделить два основных направления:

- изучение способностей к распознаванию, которыми обладают живые существа, объяснение и моделирование их;
- развитие теории и методов построения устройств, предназначенных для решения отдельных задач в прикладных целях.

Наиболее часто в задачах распознавания образов рассматриваются монохромные изображения, что дает возможность рассматривать изображение как функцию на плоскости. Если рассмотреть точечное множество на плоскости T , где функция $x(x,y)$ выражает в каждой точке изображения его характеристику — яркость, прозрачность, оптическую плотность, то такая функция есть формальная запись изображения.

Множество же всех возможных функций $x(x,y)$ на плоскости T — есть модель множества всех изображений X . Вводя понятие *сходства* между образами можно поставить задачу распознавания. Конкретный вид такой постановки сильно зависит от последующих этапов при распознавании в соответствии с тем или иным подходом.

Для оптического распознавания образов можно применить метод перебора вида объекта под различными углами, масштабами, смещениями и т. д. Для букв нужно перебирать шрифт, свойства шрифта и т. д.

Рассмотрим особенности постановки и решения задачи распознавания буквенных образов в терминах искусственных нейронных сетей (ИНС).

ИНС - это набор нейронов, соединенных между собой. Как правило, передаточные (активационные) функции всех нейронов в сети фиксированы, а веса являются параметрами сети и могут изменяться. Некоторые входы нейронов помечены как внешние входы сети, а некоторые выходы - как внешние выходы сети. Подавая любые числа на входы сети, можно получить какой-то набор чисел на выходах сети. Таким образом, работа ИНС состоит в преобразовании входного вектора X в выходной вектор Y , причем это преобразование задается весами сети.

Задача построения ИНС решается в два этапа.

1. Выбор типа (архитектуры) сети.
2. Подбор весов (обучение) сети.

На первом этапе следует выбрать следующее:

- какие нейроны мы хотим использовать (число входов, передаточные функции);
- каким образом следует соединить их между собой;
- что взять в качестве входов и выходов сети.

В настоящее время существует несколько десятков различных нейросетевых архитектур, наиболее популярными из которых являются многослойный персептрон, нейросеть с общей регрессией, сети Кохонена и другие.

На втором этапе нам следует «обучить» выбранную сеть, т.е. подобрать такие значения ее весов, чтобы сеть работала нужным образом. В используемых ИНС количество весов может составлять несколько десятков тысяч, поэтому обучение - достаточно сложный процесс. Для многих архитектур разработаны специальные алгоритмы обучения, которые позволяют настроить веса сети определенным образом.

В зависимости от функций, выполняемых нейронами в сети, можно выделить три их типа:

- входные нейроны - это нейроны, на которые подается входной вектор, кодирующий входное воздействие или образ внешней среды; в них обычно не осуществляется вычислительных процедур, информация передается с входа на выход нейрона путем изменения его активации;
- выходные нейроны - это нейроны, выходные значения которых представляют выход сети;
- промежуточные нейроны - это нейроны, составляющие основу искусственных нейронных сетей.

В большинстве нейронных моделей тип нейрона связан с его расположением в сети. Если нейрон имеет только выходные связи, то это входной нейрон, если наоборот - выходной нейрон. Однако может встретиться случай, когда выход топологически внутреннего нейрона рассматривается как часть выхода сети. В процессе функционирования сети осуществляется преобразование входного вектора в выходной,

т.е. некоторая переработка информации. Конкретный вид выполняемого сетью преобразования информации обуславливается не только характеристиками нейроподобных элементов (функциями активации и т.п.), но и особенностями ее архитектуры, а именно, той или иной топологией межнейронных связей, выбором определенных подмножеств нейроподобных элементов для ввода и вывода информации, способами обучения сети, наличием или отсутствием конкуренции между нейронами, направлением и способами управления и синхронизации передачи информации между нейронами.

С точки зрения топологии, среди нейронных сетей, сформированных на основе нейроподобных элементов, можно выделить три основных типа (рис. 1):

- полносвязные сети;
- многослойные или слоистые сети;
- слабосвязные сети (нейронные сети с локальными связями).

Полносвязные сети представляют собой ИНС, каждый нейрон которой передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

В многослойных сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. В общем случае сеть состоит из Q слоев, пронумерованных слева направо.

Внешние входные сигналы подаются на входы нейронов первого слоя (входной слой часто нумеруют как нулевой), а выходами сети являются выходные сигналы последнего слоя. Вход нейронной сети можно рассматривать как выход «нулевого слоя» вырожденных нейронов, которые служат лишь в качестве распределительных точек, суммирования и преобразования сигналов здесь не производится.

Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько промежуточных (скрытых) слоев. Связи от выходов нейронов некоторого слоя q к входам нейронов следующего слоя $(q+1)$ называются последовательными. В свою очередь, среди слоистых сетей выделяют следующие типы.

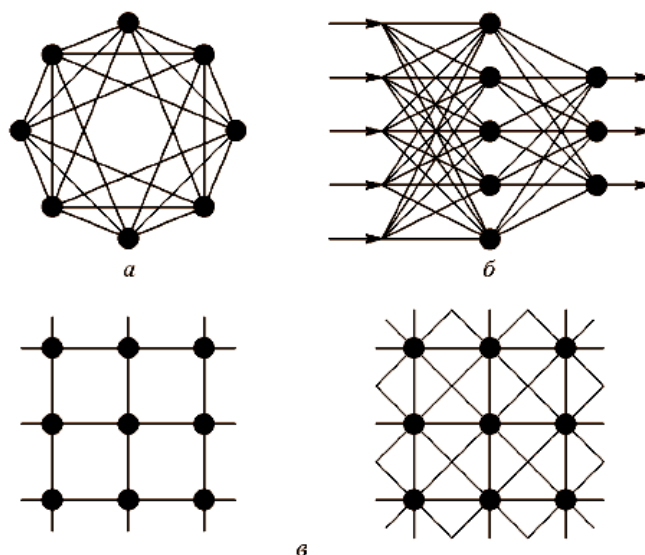


Рис. 1. Архитектуры нейронных сетей:

а — полносвязная сеть; б — многослойная сеть с последовательными связями; в — слабосвязные сети.

1. *Монотонные.* Это специальный частный случай слоистых сетей с дополнительными условиями на связи и элементы. Каждый слой, кроме последнего (выходного), разбит на два блока: возбуждающий (В) и тормозящий (Т). Связи между блоками тоже разделяются на тормозящие и возбуждающие. Для элементов монотонных сетей необходима монотонная зависимость выходного сигнала элемента от параметров входных сигналов.

2. *Сети без обратных связей.* В таких сетях нейроны входного слоя получают входные сигналы, преобразуют их и передают нейронам 1-го скрытого слоя, далее срабатывает 1-й скрытый слой и т.д. до Q -го, который выдает выходные сигналы для интерпретатора и пользователя. Если не оговорено противное, то каждый выходной сигнал q -го слоя подается на вход всех нейронов $(q+1)$ -го слоя; однако возможен вариант соединения q -го слоя с произвольным $(q+p)$ слоем. Классическим вариантом слоистых сетей являются сети прямого распространения (рис. 2).

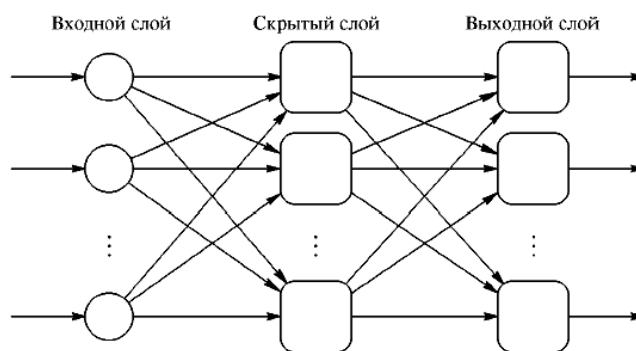


Рис. 2. Многослойная (двухслойная) сеть прямого распространения

3. *Сети с обратными связями.* Это сети, у которых информация с последующих слоев передается на предыдущие. В качестве примера сетей с обратными связями на рис. 2.3 представлены так называемые частично-рекуррентные сети Элмана и Жордана.

Известные сети можно разделить по принципу структуры нейронов в них на гомогенные или однородные и гетерогенные. Гомогенные сети состоят из нейронов одного типа с единой функцией активации. В гетерогенную сеть входят нейроны с различными функциями активации.

ИНС также можно классифицировать по типу обрабатываемых сигналов на бинарные и аналоговые. Первые из них оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения: логический ноль («заторможенное» состояние) и логическая единица («возбужденное» состояние).

По природе изменения состояния нейронов во времени существуют асинхронные и синхронные ИНС. В первом случае в каждый момент времени свое состояние меняет лишь один нейрон. Во втором - состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Ход времени в ИНС задается итерационным выполнением однотипных действий над нейронами. Кроме того, ИНС можно классифицировать по числу слоев.

Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется ИНС. Чем сложнее ИНС, тем сложнее задачи, которые она может решить.

Выбор структуры ИНС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации.

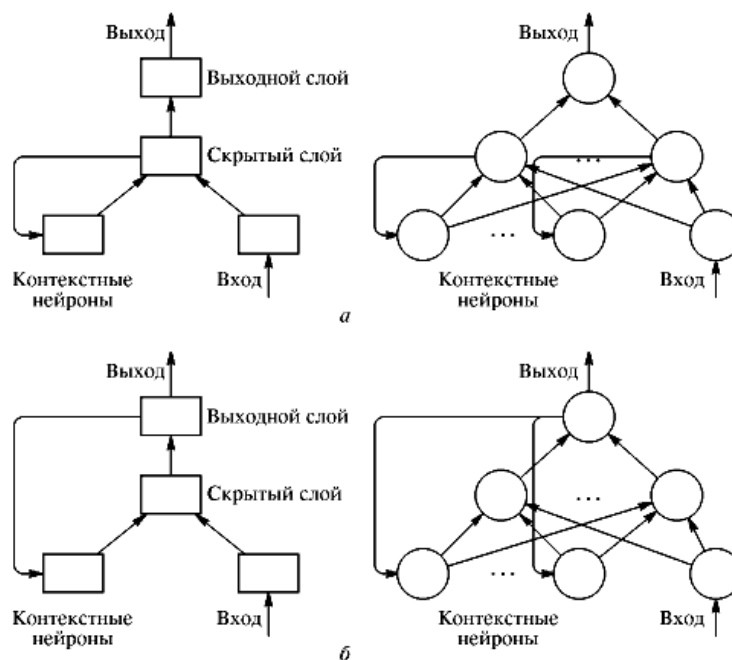


Рис. 3. Частично-рекуррентные сети: а — Элмана;
б — Жордана

Так как проблема синтеза ИНС сильно зависит от решаемой задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора, однако, существуют доказательства того, что для любого алгоритма существует нейронная сеть, которая может его реализовать.

Многие задачи: распознавания образов (зрительных, речевых и т.д.), выполнения функциональных преобразований при обработке сигналов, управления, прогнозирования, идентификации сложных систем и т.д., сводятся к следующей математической постановке. Необходимо построить отображение $X \rightarrow Y$ такое, чтобы на каждый возможный входной сигнал X формировался правильный выходной сигнал Y . Отображение задается конечным набором пар ((вход), (известный выход)). Число таких пар (обучающих примеров) существенно меньше общего числа возможных

сочетаний значений входных и выходных сигналов. Совокупность всех обучающих примеров носит название обучающей выборки.

В задачах распознавания образов X - некоторое представление образа (изображение, вектор чисел и т. д.), Y - номер класса, к которому принадлежит входной образ.

В задачах управления X - набор контролируемых параметров управляемого объекта, Y - код, определяющий управляющее воздействие, соответствующее текущим значениям контролируемых параметров.

В задачах прогнозирования в качестве входных сигналов используются временные ряды, представляющие значения контролируемых переменных на некотором интервале времени. Выходной сигнал - множество переменных, которое является подмножеством переменных входного сигнала.

При идентификации X и Y представляют входные и выходные сигналы системы соответственно.

Таким образом, большая часть прикладных задач может быть сведена к реализации некоторого сложного многомерного функционального преобразования. В результате построения такого отображения необходимо добиться того, чтобы обеспечивалось формирование правильных выходных сигналов в соответствии со всеми примерами обучающей выборки и формирование правильных выходных сигналов в соответствии со всеми возможными входными сигналами, которые не вошли в обучающую выборку. Второе требование в значительной степени усложняет задачу формирования обучающей выборки.

Для оценки числа нейронов в скрытых слоях однородных нейронных сетей можно воспользоваться формулой для оценки необходимого числа синаптических весов L_w (в многослойной сети с сигмоидальными передаточными функциями):

$$\frac{mN}{1 + \log_2 N} \leq L_w \leq m \left(\frac{N}{m} + 1 \right) (n + m + 1) + m,$$

где n - размерность входного сигнала,

m - размерность выходного сигнала,

N - число элементов обучающей выборки.

Оценив необходимое число весов, можно рассчитать число нейронов в скрытых слоях.

$$L = \frac{L_w}{n + m}.$$

$$2(L + n + m) \leq N \leq 10(L + n + m),$$
$$\frac{N}{10} - n - m \leq L \leq \frac{N}{2} - n - m.$$

Так же можно рассчитать число нейронов в сетях с большим числом слоев, которые иногда целесообразно использовать: такие многослойные нейронные сети могут иметь меньшие размерности матриц синаптических весов нейронов одного слоя, чем двухслойные сети, реализующие то же самое отображение.

Таким образом, нейронные сети являются универсальными аппроксимирующими системами.

Очевидно, что процесс функционирования ИНС, т.е. сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, задавшись определенной структурой ИНС, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

Этот этап называется обучением ИНС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время функционирования.

Общая схема процесса обучения приведена на рис. 2.4.

После многократного предъявления примеров из обучающей выборки веса сети стабилизируются, причем сеть дает правильные ответы на все (или почти все) примеры из базы данных. В таком случае говорят, что «сеть выучила все примеры», «сеть обучена», или «сеть натренирована». В программных реализациях можно видеть, что в процессе обучения функция ошибки (например, сумма квадратов ошибок по всем выходам) постепенно уменьшается. Когда функция ошибки достигает нуля или приемлемого малого уровня, обучение (тренировку) останавливают, а полученную сеть считают натренированной и готовой к применению на новых данных.

Вся информация, которую сеть имеет о задаче, содержится в наборе примеров. Поэтому качество обучения сети напрямую зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают данную задачу.



Рис.4. Общая схема процесса обучения ИНС

Математически процесс обучения можно описать следующим образом.

В процессе функционирования нейронная сеть формирует выходной сигнал Y в соответствии с входным сигналом X , реализуя некоторую функцию $Y = G(X)$. Если архитектура сети задана, то вид функции G определяется значениями синаптических весов и смещений сети.

Пусть решением некоторой задачи является функция $Y = F(X)$, заданная парами входных-выходных данных $(X^1, Y^1), (X^2, Y^2), \dots, (X^N, Y^N)$, для которых $Y^k = F(X^k)$ ($k = 1, 2, \dots, N$). Обучение состоит в поиске (синтезе) функции G , близкой к F в смысле некоторой функции ошибки E .

Если выбраны множество обучающих примеров - пар (X^k, Y^k) (где $k = 1, 2, \dots, N$) и способ вычисления функции ошибки E , то обучение нейронной сети превращается в задачу многомерной оптимизации, имеющую очень большую размерность, при этом, поскольку функция E может иметь произвольный вид, обучение в общем случае - многоэкстремальная невыпуклая задача оптимизации.

Для решения этой задачи могут быть использованы следующие (итерационные) алгоритмы:

- алгоритмы локальной оптимизации с вычислением частных производных первого порядка;
- алгоритмы локальной оптимизации с вычислением частных производных первого и второго порядка;
- стохастические алгоритмы оптимизации;
- алгоритмы глобальной оптимизации.

К первой группе относятся: градиентный алгоритм (метод скорейшего спуска); методы с одномерной и двумерной оптимизацией целевой функции в направлении антиградиента; метод сопряженных градиентов; методы, учитывающие направление антиградиента на нескольких шагах алгоритма.

Ко второй группе относятся: метод Ньютона, методы оптимизации с разреженными матрицами Гессе, квазиньютоновские методы, метод Гаусса-Ньютона, метод Левенберга-Марквардта и др.

Стохастическими методами являются: поиск в случайном направлении, имитация отжига, метод Монте-Карло (численный метод статистических испытаний).

Задачи глобальной оптимизации решаются с помощью перебора значений переменных, от которых зависит целевая функция (функция ошибки E).

В простейшем случае задача распознавания 2 буквенных формулируется следующим образом:

Дано: растровое черно-белое изображение буквы размером 5×5 элементов. Необходимо определить какая это буква.

В терминах нейронных сетей данная задача формулируется следующим образом:

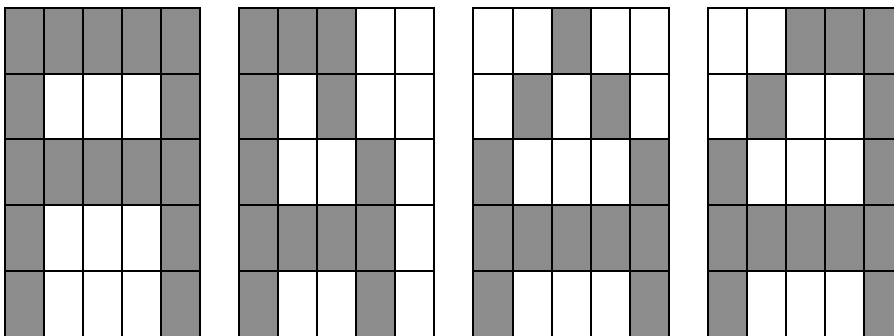
Дано: Входной вектор из 25 двоичных символов $X=(x_1, x_2, \dots, x_{25})$. Необходимо построить искусственную нейронную сеть с 25 входами, и 2 выходами, которые помечены буквами. Если на выходе сети – изображение буквы «А», то максимальное значение выходного сигнала достигается на выходе 1.

Уровень выходного сигнала может принимать любые значения. Однако в данной задаче основное внимание уделяется не аналоговому ответу, а номеру класса (номеру буквы). Поэтому каждому классу (букве) сопоставляется свой выход, а от-

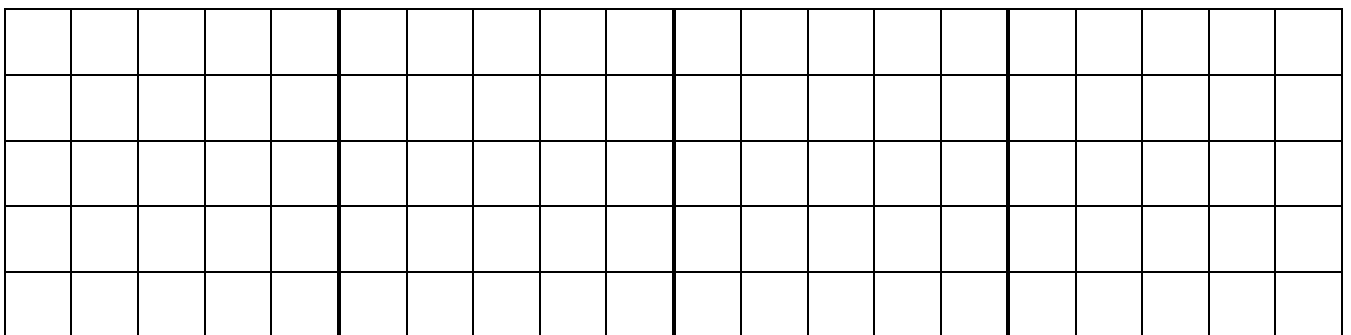
ветом сети считается тот класс (буква), но чьем выходе уровень сигнала максимальный, т.е. уровень сигнала на выходе «А» - это достоверность того, что на вход подана буква «А».

Для формирования обучающей выборки составляется таблица, в которой имеется 25 столбцов, соответствующих элементам входного вектора ($x_1 \dots x_{25}$) и 2 столбца, соответствующих элементам выходного вектора (y_1, y_2). Число строк соответствует числу примеров обучающей выборки.

Рассмотрим правило получения обучающих примеров для буквы «А»

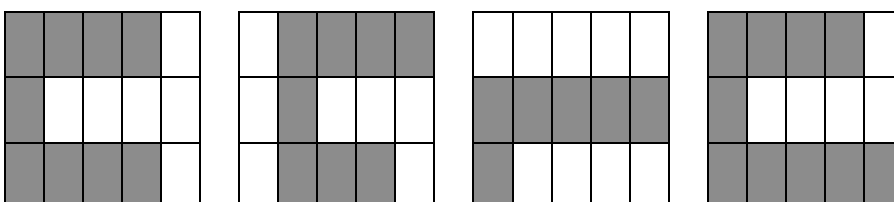


Данные буквенные образы в случае монохромного представления кодируются следующим образом:



Для изображений соответствующих букве «А», $y_1=1, y_2=0$.

Аналогично кодируются буквенные образы для буквы «Б».



образов в соответствии с вариантом из таблицы, состоящую примерно из 5-10 обучающих примеров для каждой буквы одним из способов доступных для программ NEUROPRO и NEURAL ANALYZER (Таблица 1).

3. Использую навыки решения задач прогнозирования и классификации с использованием программ NEUROPRO и NEURAL ANALYZER. осуществить подключение к источнику данных, содержащем обучающую выборку и произвести следующие действия:

- создать новое подключение с соответствующим именем и параметрами. При необходимости изменить названия меток поля, просмотреть данные.
- настроить поля обучающей выборки в соответствии с заданием, установить параметры нормализации данных.

Варианты заданий

№ варианта	Буквы	
1	А	Б
2	В	Г
3	И	К
4	Л	Н
5	О	П
6	Р	С
7	Т	Х
8	Н	О
9	Е	Н
10	Р	Г

- провести обучение НС по различным алгоритмам обучения с различными параметрами обучения и условиями остановки.
- проконтролировать результаты обучения НС, используя окно эксперимента и различные методики тестирования ИНС.

- сформировать рабочую выборку различными способами (заполнение вручную и импорт данных), рассчитать выходные поля рабочей выборки и проанализировать статистические характеристики рабочей выборки для входных и выходных полей.
- построить различные графики выходов сети, проанализировать полученные результаты.
- провести сравнительный анализ качества решения задачи распознавания буквенных образов с помощью разных программных средств нейросетевого моделирования.
- сделать выводы о влиянии структуры и параметров сети на скорость обучения НС и качество решения конкретной задачи распознавания образов.

ЛАБОРАТОРНАЯ РАБОТА №5

РЕШЕНИЕ НЕФОРМАЛИЗОВАННЫХ ЗАДАЧ ПРОГНОЗИРОВАНИЯ И КЛАССИФИКАЦИИ НА ОСНОВЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ОБРАБОТКИ ИНФОРМАЦИИ

Цель работы: изучение математических моделей и алгоритмов функционирования искусственных нейронных сетей (ИНС), получение практических навыков по проектированию, обучению, анализу и оптимизации ИНС в пакете прикладных программ NeuroPro.

Под ИНС понимается некоторое вычислительное устройство обработки информации, состоящее из большого числа параллельно работающих простых процессорных элементов – нейронов, связанных между собой линиями передачи информации – связями или синапсами. У ИНС выделена группа связей, по которым она получает информацию из внешнего мира, и группа выходных связей, с которых снимаются выдаваемые сетью сигналы. ИНС применяются для решения различных задач классификации и прогнозирования. ИНС обучается решению задачи на основании некоторой обучающей выборки – "задачника", состоящего из набора пар "вход–требуемый выход", и далее способна решать примеры, не входящие в обучающую выборку

Среди всего множества нейросетевых архитектур можно выделить две базовых архитектуры – *слоистые* и *полносвязные* сети.

Слоистые сети: нейроны расположены в несколько слоев. Нейроны первого слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до k -го слоя, который выдает выходные сигналы. Если не оговорено противное, то каждый выходной сигнал i -го слоя подается на вход всех нейронов $i+1$ -го. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: каждый нейрон первого слоя получает все входные сигналы.

Можно выделить два класса задач, решаемых обучаемыми нейронными сетями. Это задачи предсказания и классификации. Задачи предсказания или прогнозирования являются, задачами построения регрессионной зависимости выходных данных от входных. ИНС могут эффективно строить сильно нелинейные регрессионные зависимости.

На веса синапсов сети обычно наложены требования принадлежности некоторому диапазону значений. Наиболее часто используемые нелинейные функции нейронов также обычно выдают значения из некоторого диапазона. Это приводит к тому, что обычно нельзя подавать сети входные сигналы в их истинном диапазоне величин и получать от сети выходные сигналы в требуемом диапазоне. Поэтому перед подачей сети входных сигналов их необходимо нормировать, например, в диапазон значений $[-1,1]$ или $[0,1]$ в тех случаях, когда существенна положительность сигнала, либо делать так, чтобы входные сигналы не слишком сильно выходили за пределы этих отрезков. При нормировке каждая компонента входного вектора данных x_i заменяется величиной

$$x_i = \frac{x_i - (\max x_i + \min x_i)/2}{(\max x_i - \min x_i)/2},$$

где $\max x_i$ и $\min x_i$ – соответственно максимальное и минимальное значение для данной компоненты, вычисленные по всей обучающей выборке. По этой же формуле пересчитываются и компоненты векторов ответов.

Обучение нейронной сети означает минимизацию функционала невязки между выходными сигналами сети (действительными выходами) и сигналами, которые требуется получить (желаемыми выходами). Минимизация производится путем такой подстройки обучаемых параметров нейронов сети (весовых коэффициентов синапсов), чтобы сеть на некоторый входной вектор сигналов X выдавала такой ответ Y' , который был бы по возможности ближе к требуемому ответу Y .

После обучения нейронной сети необходимо провести ее тестирование на тестовой выборке для определения точности решения не входивших в обучающую выборку задач. Точность правильного решения очень сильно зависит от репрезентативности обучающей выборки.

Упрощение (контрастирование) нейронной сети строится как последовательный процесс исключения из сети наименее значимого элемента и дальнейшего обучения ИНС.

После того, как сеть обучилась решению некоторой задачи, обычно не ясно, какие же правила используются сетью для получения правильных ответов. ИНС производит из задачника (таблицы данных) скрытое знание – формирует некоторый навык прогнозирования или классификации, но логическая структура этого навыка осталась непонятной пользователю. Этот факт называется логической непрозрачностью ИНС. Логически прозрачной называется ИНС у которой легко можно сформировать понятный пользователю алгоритм решения задачи.

После того, как сеть приведена к логически прозрачному виду, необходимо вербализовать навык, заложенный в ИНС, в понятном пользователю

виде. Это делается на основе графического визуального либо текстового представления структуры нейронной сети. Вербализация проводится как процесс восстановления симптом-синдромной структуры понятий предметной области

Программа NeuroPro представляет собой менеджер обучаемых искусственных нейронных сетей, работающий в среде операционной системы Windows и позволяющий производить следующие базовые операции:

1. Создание нейропроекта;
2. Подключение к нейропроекту файла (базы) данных в формате dfb (dBase, FoxBase, FoxPro, Clipper) или db (Paradox);
3. Редактирование файла данных – изменение существующих значений и добавление новых записей в базу данных; сохранение файла данных в другом формате;
4. Добавление в проект нейронной сети слоистой архитектуры с числом слоев нейронов от 1 до 10, числом нейронов в слое – до 100;
5. Обучение нейронной сети решению задачи прогнозирования или классификации. ИНС может одновременно решать как несколько задач прогнозирования (прогнозирование нескольких чисел), так и несколько задач классификации, а также одновременно задач и прогнозирования, и классификации.
6. Тестирование нейронной сети на файле данных, получение статистической информации о точности решения задачи;
7. Вычисление показателей значимости входных сигналов сети, сохранение значений показателей значимости в текстовом файле на диске;
8. Упрощение нейронной сети;
9. Генерация и визуализация вербального описания нейронной сети, сохранение вербального описания в текстовом файле на диске;
10. Выбор алгоритма обучения, назначение требуемой точности прогноза, настройка нейронной сети.

В качестве файлов данных (содержащих обучающую выборку для нейронных сетей) используются файлы форматов DBF (форматы пакетов Dbase, FoxBase, FoxPro, Clipper) и DB (Paradox). Возможно чтение и редактирование этих файлов и сохранение измененных файлов на диске. Программа не накладывает ограничений на число записей (строк) в файле данных.

Файлы нейропроекта имеют уникальный формат, поддерживаемый только настоящей программой. Файлы вербального описания сети, результатов тестирования нейросети, показателей значимости входных сигналов имеют стандартный ASCII-формат текстовых файлов данных и могут читаться всеми программами-редакторами текстов и импортироваться в электронные таблицы.

В данной программе имеется возможность работы только со слоистыми и слоистыми монотонными нейронными сетями с числом слоев нейронов от 1 до 10, числом нейронов в слое – до 100. Число нейронов в слое не зависит от числа входных сигналов и числа выходных сигналов. После последнего слоя нейронов сеть имеет слой адаптивных сумматоров с числом сумматоров, равных числу выходных сигналов, с которых и снимаются выходные сигналы сети.

Число входных и выходных сигналов сети ограничено максимальным числом полей в файле данных и не может в сумме превышать 255 или 511 (в зависимости от файла данных) сигналов. Каждому входному и выходному сигналу соответствует поле в файле данных. Веса синапсов при обучении могут изменяться в диапазоне $[-1,1]$, при создании сети инициализируются случайными числами.

Нелинейная функция активации нейрона имеет вид $f(A)=A/(c+|A|)$, где c - параметр крутизны переходного участка сигмоидной функции, который задается при создании сети в диапазоне $[0.0001,1]$ и не изменяется при обучении. Параметр крутизны можно задавать отдельно для каждого слоя сети.

Обрабатываемые поля могут быть непрерывными (количественными) и дискретнозначными (качественными).

Каждое количественное поле в файле данных будет соответствовать одному входному или выходному сигналу сети. Перед подачей количественных входных полей нейронной сети происходит их нормировка в диапазон значений $[-1,1]$ по каждому полю. Выходные сигналы сети нормируются в диапазон истинных значений.

Для прогнозируемого качественного признака точность означает максимально допустимое отклонение прогноза сети от истинного значения признака. Желательно задавать как можно менее жесткие требования к точности. Это ускорит как процесс обучения, так и процесс упрощения сети. Также задачу можно будет решить на основе нейронной сети с меньшим числом слоев или нейронов, и, обычно, на основании меньшего числа входных сигналов.

Обучение нейронной сети на некотором задачнике производится градиентными методами оптимизации, градиент вычисляется по принципу двойственности. В программе реализованы четыре алгоритма оптимизации:

- Градиентный спуск.
- Модифицированный ParTan.
- Метод сопряженных градиентов.
- Квазиньютоновский BFGS-метод.

При создании нейропроекта в качестве алгоритма по умолчанию принимается ParTan.

Примеру задачника соответствует запись (строка) файла данных. Для включения записи файла данных в задачник в записи должны присутствовать данные для всех полей, используемых нейронной сетью в качестве входных и выходных.

Обучение прекращается при достижении заданной точности решения задачи либо при невозможности дальнейшей оптимизации.

В NeuroPro предоставляются следующие возможности по упрощению сети:

- сокращение числа входных сигналов;

- сокращение числа нейронов сети;
- сокращение числа синапсов сети;
- сокращение числа неоднородных (пороговых) входов нейронов сети.
- равномерное упрощение сети, чтобы на каждый нейрон сети приходило не более n сигналов.
- бинаризация весов синапсов и неоднородных входов сети.

Сокращение входных сигналов и нейронов может достигаться и при выполнении других операций по упрощению сети, а не только при целенаправленном сокращении именно входных сигналов и нейронов.

Нейрон сети считается удаленным, если у него удалены все выходы или его сигнал не используется нейронами следующего слоя.

Входной сигнал считается удаленным, если удалены все синапсы, по которым этот сигнал поступал на нейроны первого слоя сети.

Удаленные при упрощении элементы физически остаются в нейронной сети, но при генерации вербального описания сети не вносятся в вербальное описание.

При генерации вербального описания в тексте перечисляются используемые поля файла данных, правила их предобработки для подачи сети, описание нелинейных функций нейронов, функционирование нейронной сети послойно и понейронно, правила нормировки выходных сигналов сети в диапазон истинных значений. Сигналам, генерируемым нейронами сети, присваиваются некоторые имена и в дальнейшем пользователь при анализе сети может именовать эти сигналы в терминах проблемной области.

Получив вербализованное описание нейронной сети, пользователь может попытаться восстановить правила, сформированные сетью для решения задачи – записать на естественном языке алгоритм решения неформализованной задачи предсказания или классификации.

Работа с нейронными сетями возможна только в рамках некоторого проекта. Для того, чтобы создать проект, необходимо выбрать пункт меню *Файл->Создать* или нажать кнопку "Создать" в панели кнопок. После создания нейропроекта в окне нейропроекта в него можно вставлять ИНС и работать с последними. Созданный нейропроект может быть сохранен на диске при помощи команд меню *Файл->Сохранить*, *Файл->Сохранить как* или нажатием на кнопку "Сохранить".

В дальнейшем возможна работа с сохраненными файлами нейропроекта – для этого необходимо выбрать пункт меню *Файл->Открыть* или нажать кнопку "Открыть" и далее выбрать в диалоговом окне имя нужного нейропроекта.

Окно нейропроекта (рис. 1) содержит следующие элементы:

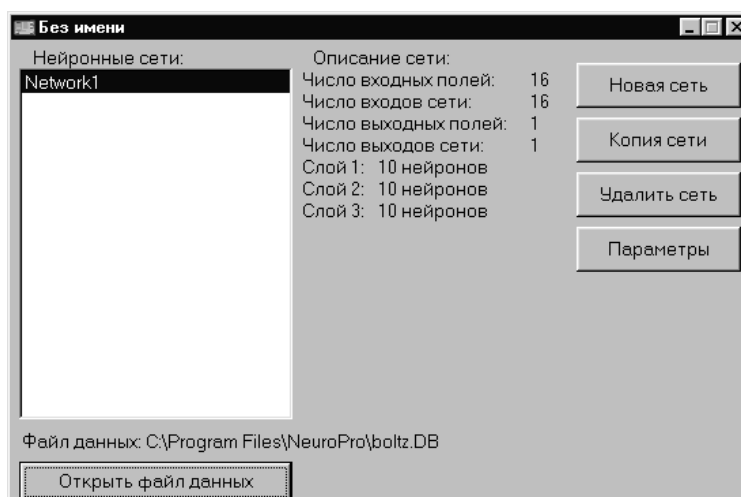


Рис.1. Окно нейропроекта

ИНС – список имеющихся в проекте нейронных сетей. Работа ведется с выбранной в настоящий момент нейронной сетью. Перемещение по списку изменяет активную нейронную сеть.

Описание сети – информация о структуре активной нейронной сети – числе слоев и нейронов в сети, числе входных и выходных сигналов. Эта информация отражает начальную структуру.

Файл данных – имя подключенного к проекту файла данных.

Открыть файл данных – вызов диалога открытия файла для смены подключенного к проекту файла данных.

Новая сеть – вызов диалога создания новой нейронной сети. *Копия сети* – создание новой ИНС, по структуре совпадающей с текущей.

Удалить сеть – удаление активной нейронной сети из проекта.

Параметры – диалог смены параметров активной нейронной сети, который позволяет менять только точность решения задачи и параметры крутизны характеристики нейронов.

Большинство операций с нейронными сетями требуют присутствия подключенного к нейропроекту файла данных. Для подключения файла данных или его замены необходимо нажать кнопку *Открыть файл данных* в окне нейропроекта и далее выбрать имя необходимого файла данных. Открытый файл данных отображается в собственном окне, где предоставляется возможность его редактирования.

ИНС, созданные для этого файла данных, обучаются только на том множестве строк файла, где для входных и выходных полей имеются все данные. Тестирование нейронных сетей происходит по тем строкам файла, где имеются все данные во входных полях. При закрытии окна файла данных подключение к нейропроекту завершается.

При подключенном файле данных можно проводить операции создания новых нейросетей, их обучения, тестирования и упрощения.

Для создания новой нейронной сети необходимо нажать кнопку *Новая сеть* в окне нейропроекта и заполнить *Диалог создания нейронной сети*. Созданную нейронную сеть можно далее обучать, тестировать, упрощать и сохранять на диске вместе с нейропроектом.

Поле является выходным для сети – ИНС обучается прогнозировать значения этого поля.

Входы и выходы – лист для определения использования нейронной сетью имеющихся в файле данных полей (рис.2).

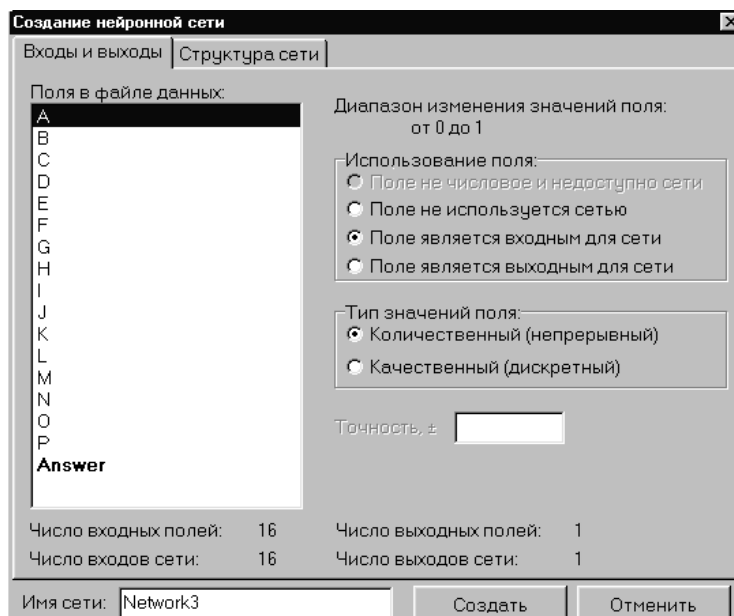


Рис.2. Диалог создания нейронной сети (входы и выходы)

Поля в файле данных – список полей в файле данных.

Использование поля – использование текущего поля нейронной сетью.

Поле не числовое и недоступно сети – поле не является числовым и не может обрабатываться нейронной сетью.

Поле не используется сетью – данное числовое поле не используется сетью.

Поле является входным для сети – значения данного поля подаются на входы сети.

Тип значений поля – определение значений, которые принимает текущий признак:

Количественный (непрерывный) – признак принимает значения из некоторого непрерывного диапазона значений.

Качественный (дискретный) – признак принимает значения из конечного набора дискретных значений. Для определения набора дискретных состояний появляется лист Дискретные состояния (см. далее).

Диапазон изменения значений поля – минимальное и максимальное значение поля в файле данных.

Надежность или *Точность* – требования к точности решения задачи.

Для количественного признака точность может изменяться от 0 (максимальная точность, не должно быть отличий прогноза сети от известного значения) до ширины диапазона изменения значений этого поля (минимальная точность). По умолчанию предлагается точность в 10% от ширины диапазона, при этом сеть должна обучиться предсказывать значения данного поля с точностью $\pm 10\%$ от ширины диапазона изменения значений. Чем меньше величина допустимой точности, тем более точно сеть должна научиться предсказывать известные значения.

Число входных полей – число полей в файле данных, используемых сетью в качестве входных.

Число входов сети – число входных сигналов сети.

Число выходных полей – число полей в файле данных, используемых сетью в качестве выходных.

Число выходов сети – число выходных сигналов сети.

Структура сети – лист для задания структуры нейронной сети (рис. 3).

Создание нейронной сети

Входы и выходы | Структура сети

Число слоев нейронов: 3

Монотонность: ☐

	Число нейронов:	Нелинейность:	Характеристика:
Слой 1:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 2:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 3:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 4:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 5:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 6:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 7:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 8:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 9:	10	Сигмоида $f(X)=X/(c+ X)$	0.1
Слой 10:	10	Сигмоида $f(X)=X/(c+ X)$	0.1

Имя сети: Network3

Создать Отменить

Рис. 3. Диалог создания нейронной сети (структура сети)

Монотонность – создание сети монотонной архитектуры.

Число слоев нейронов – число слоев нейронов в сети. Изменяется от 1 до 10. По умолчанию предлагается 3 слоя нейронов. Для каждого слоя нейронов возможно задание следующих характеристик:

Число нейронов – число нейронов в слое. Изменяется от 1 до 100. По умолчанию предлагается 10 нейронов в слое.

Нелинейность – вид нелинейного преобразователя нейронов данного слоя - сигмоидная функция вида $f(A)=A/(c+|A|)$, где c - характеристика нейрона.

Характеристика – значение необучаемой константы, используемой нелинейным преобразователем. Может изменяться в диапазоне от 0,0001 до 1 для описанной выше сигмоидной нелинейности. По умолчанию предлагается значение характеристики, равное 0,1.

Имя сети – имя нейронной сети в списке сетей нейропроекта.

Создать/Изменение – создание нейронной сети или внесение изменений в ее параметры.

Отменить – отмена создания или изменения параметров нейронной сети.

После создания нейронной сети она появляется в списке сетей нейропроекта и становится активной.

Для обучения активной в данный момент в нейропроекте нейронной сети необходимо выбрать пункт меню *ИНС/Обучение*. При этом на экран выводится *Окно обучения и упрощения сети*, где пользователь имеет возможность наблюдать процесс обучения и при необходимости самостоятельно завершить обучение нажатием кнопки *Завершить*.

В окне обучения и упрощения нейронной сети отображаются следующие характеристики (рис.4.).

Число циклов обучения – общее число шагов обучения (шагов градиентного спуска, ParTan-шагов, шагов метода сопряженных градиентов или BFGS-метода).

Шаг – величина шага в направлении оптимизации. При невозможности оптимизации в этом поле отображается строка *Zero step* или *Infinite step* – соответственно при нулевом или бесконечном шаге в направлении оптимизации.

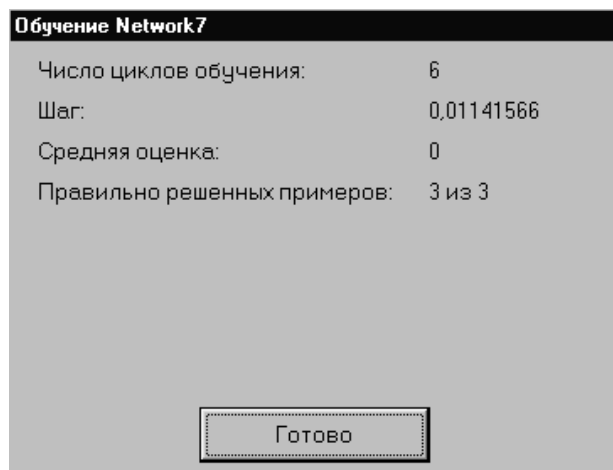


Рис. 4. Окно обучения и упрощения сети

Средняя оценка – средняя оценка на обучающем множестве.

Правильно решенных примеров – число примеров с нулевой оценкой из общего числа примеров, по которым производится обучение сети.

Удалено входов – число удаленных входных сигналов сети.

Удалено нейронов – число удаленных у сети нейронов.

Удалено синапсов – число удаленных синапсов и неоднородных входов нейронов сети.

Бинаризовано синапсов – число бинаризованных синапсов и неоднородных входов нейронов сети.

Завершить/Готово – кнопка прекращения процесса обучения или упрощения сети.

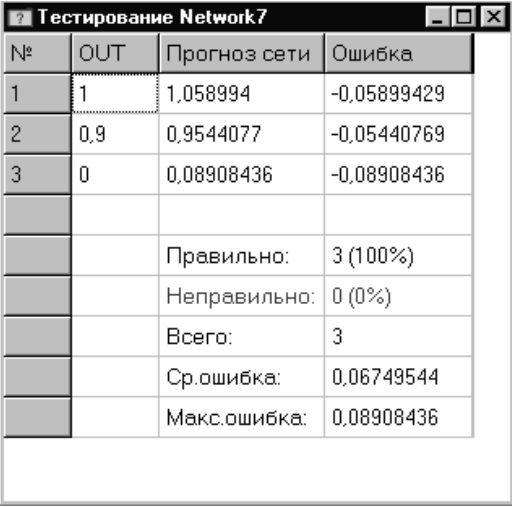
Обучение прекращается автоматически при достижении нулевого значения средней оценки на задачнике, в случае невозможности дальнейшего улучшения оценки либо при аварийных ситуациях (нулевой или бесконечный шаг в направлении оптимизации).

Упрощение прекращается автоматически при невозможности достижения нулевого значения средней оценки после очередного акта упрощения сети. При этом сеть возвращается к предыдущему состоянию полной обученности.

Имея нейронную сеть, можно посмотреть, насколько точно она прогнозирует значения выходных полей в файле данных. Для тестирования нейронной сети необходимо выбрать пункт меню *ИНС/Тестирование*. Результат тестирования сети выводится в *Окно тестирования сети*.

В окне тестирования сети (рис. 5) выводятся значения полей файла данных, которые являются выходными для нейронной сети, и значения прогноза этих полей нейронной сетью. Если для какой-либо строки в файле неизвестно значение выходного поля, то для этой записи будет выведен только прогноз сети.

Содержимое окна прогноза можно сохранить в текстовом файле для последующей обработки в другой программе (например, Microsoft Excel).



№	OUT	Прогноз сети	Ошибка
1	1	1,058994	-0,05899429
2	0,9	0,9544077	-0,05440769
3	0	0,08908436	-0,08908436
		Правильно:	3 (100%)
		Неправильно:	0 (0%)
		Всего:	3
		Ср.ошибка:	0,06749544
		Макс.ошибка:	0,08908436

Рис. 5. Окно тестирования сети

Для каждого выходного поля информация выводится в три колонки:

1. "*Значение*" – значения исходных данных для этого поля. Для количественных данных выводятся собственно значения данных, для качественных – принадлежность их к тому или иному классу.

2. *"Прогноз"* – прогноз сети для этого поля. Для количественных данных выводятся собственно значения данных, для качественных – принадлежность их к тому или иному классу.

3. *"Уверенность"* или *"Ошибка"* – невязка между исходным и спрогнозированным значением (*"Ошибка"*).

Информация в столбцах *"Прогноз"* и *"Уверенность"/"Ошибка"* выделяется следующими цветами:

Черный – исходные значения (для поля исходных значений) и правильный ответ.

Синий – правильный ответ сети, однако не со 100% уверенностью (при решении задачи классификации).

Красный – неправильный ответ сети (при решении задачи классификации) либо отличие прогноза сети более чем на требуемое значения точности от исходного значения (для задачи прогнозирования).

Зеленый – прогноз отсутствующего в файле данных значения.

В конце таблицы для каждого поля выводится статистика правильности решения:

"Правильно" – число правильных прогнозов (и % от общего числа строк таблицы, для которых возможен прогноз этого поля).

"Неуверенно" – число правильных, но не 100%-уверенных прогнозов (и % от общего числа строк таблицы, для которых возможен прогноз этого поля) – только для задач классификации.

"Неправильно" – число неправильных прогнозов (и % от общего числа строк таблицы, для которых возможен прогноз этого поля).

«Всего» – общее число сделанных сетью прогнозов значений для этого поля.

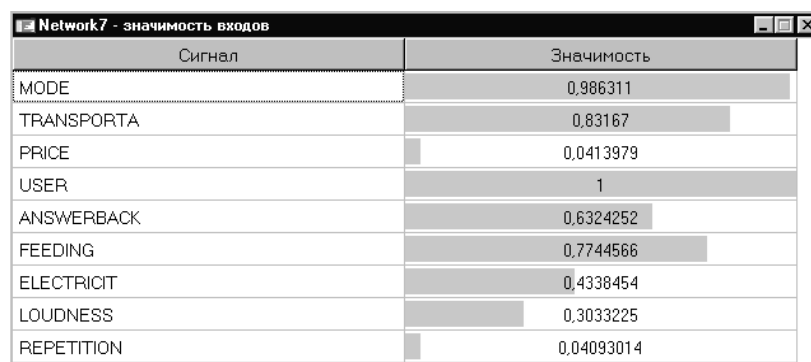
При решении задачи прогнозирования дополнительно выводятся следующие показатели:

"Ср.ошибка" – средняя (по модулю) ошибка прогноза.

"Макс.ошибка" – максимальная (по модулю) ошибка прогноза.

Возможно тестирование сети на другом файле данных. Для этого необходимо сначала подключить к проекту другой файл данных, а затем протестировать сеть.

Для вычисления показателей значимости входных сигналов нейронной сети необходимо выбрать пункт меню *ИНС/Значимость входов*. Вычисленные показатели значимости выводятся в *Окно значимости входных сигналов сети* (рис. 6.).



Сигнал	Значимость
MODE	0,986311
TRANSPORTA	0,83167
PRICE	0,0413979
USER	1
ANSWERBACK	0,6324252
FEEDING	0,7744566
ELECTRICIT	0,4338454
LOUDNESS	0,3033225
REPETITION	0,04093014

Рис. 6. Окно значимости входных сигналов сети

Окно значимости входов сети отображает текущую значимость входных сигналов для принятия сетью правильного решения. Значимость каждого из входов также отображается и в виде отнормированного на единицу числа. Для удаленных в ходе упрощения входных сигналов значимость равна 0.

Если окно значимости входов отображено на экране и для данной нейронной сети запускается процесс упрощения, то окно значимости динамически изменяет отображаемые данные после каждого перерасчета показателей значимости входных сигналов.

Не все входные сигналы сети и синапсы нейронов необходимы для правильного решения сетью задачи. Часто можно достаточно сильно упростить сеть без ухудшения точности решения задачи.

Для упрощения нейронной сети имеются следующие операции в меню *ИНС*:

Сокращение числа входных сигналов – удаление наименее значимых входных сигналов.

Сокращение числа нейронов – удаление наименее значимых нейронов сети.

Сокращение числа синапсов – удаление наименее значимых синапсов сети.

Сокращение числа неоднородных входов – удаление наименее значимых неоднородных входов нейронов сети.

Равномерное упрощение сети – сокращение числа приходящих на нейроны сети сигналов до задаваемого пользователем. Диалог равномерного упрощения сети в этом случае содержит следующие элементы: *Максимальное число сигналов на нейрон* – максимальное число сигналов, приходящих на нейрон. У каждого нейрона сети последовательно будут контрастироваться наименее значимые синапсы до тех пор, пока число оставшихся у нейрона синапсов не будет превышать заданного числа. По умолчанию максимальное число сигналов на нейрон принимается равным 3. *Упрощение* – запускает процесс равномерного упрощения сети с заданным максимальным числом сигналов. *Отменить* – отменяет запуск процесса упрощения.

Бинаризация синапсов сети – приведение значений весов синапсов и неоднородных входов нейронов к выделенным значениям.

Упрощение нейронной сети проводится до тех пор, пока возможно обучение нейронной сети до нулевой средней оценки. Текущая информация выводится в Окно обучения и упрощения сети. Упрощение может прекратиться, когда уже все синапсы, подлежащие упрощению или бинаризации, соответственно удалены или бинаризованы.

Для получения вербального описания текущей нейронной сети необходимо выбрать пункт меню *ИНС/Вербализация*. Вербальное описание сети выводится в *Окно вербального описания сети*.

Окно вербального описания сети содержит текст, описывающий нейронную сеть. Отконтрастированные входные сигналы, синапсы и нейроны сети в тексте не описываются. В текст включены следующие разделы:

Поля базы данных (исходные симптомы) – имена полей файла данных, которые используются сетью в качестве входных.

Поля базы данных (конечные синдромы) – имена полей базы данных, значения которых прогнозирует ИНС.

Предобработка входных полей БД для подачи сети – правила нормирования входных сигналов в диапазон $[-1,1]$ для подачи нейронной сети.

Функциональные преобразователи – описания используемых на каждом слое сети функциональных преобразователей нейронов.

Синдромы первого уровня – описание преобразования входных сигналов сети нейронами первого слоя сети.

Синдромы второго уровня – описание преобразования входных сигналов сети нейронами второго слоя сети (если у сети два и более слоев нейронов).

Конечные синдромы – описание вычисления значений прогнозируемых полей файла данных.

Постобработка конечных синдромов – правила нормирования количественных выходных сигналов сети из диапазона $[-1,1]$ в диапазон истинных значений.

Ниже приведен пример вербального описания нейросети.

---ВХОДЫ НЕЙРОСЕТИ-----

Поля базы данных (исходные симптомы):

X

Y

---ВЫХОД НЕЙРОСЕТИ-----

Поля базы данных (конечные синдромы):

Z

---НОРМАЛИЗАЦИЯ ВХОДОВ НЕЙРОСЕТИ-----

Предобработка входных полей БД для подачи сети:

$X=(X-0,5)/0,5$

$Y=(Y-0,5)/0,5$

ПЕРЕДАТОЧНЫЕ ФУНКЦИИ НЕЙРОНОВ 1-го 2-го 3-го СЛОЕВ

Функциональные преобразователи:

Сигмоида1(A)=A/(0,1+|A|)

Сигмоида2(A)=A/(0,1+|A|)

Сигмоида3(A)=A/(0,1+|A|)

---ВЫХОДЫ НЕЙРОНОВ 1-го СЛОЯ-----

Синдромы 1-го уровня:

Синдром1_1=Сигмоида1(0,1*X+0,4*Y+0,4)

Синдром1_2=Сигмоида1(0,2*X+0,2*Y-0,1)

---ВЫХОДЫ НЕЙРОНОВ 2-го СЛОЯ-----

Синдромы 2-го уровня:

Синдром2_1=Сигмоида2(0,2*Синдром1_2-0,1)

Синдром2_2=Сигмоида2(0,4*Синдром1_1+0,1)

Синдром2_3=Сигмоида2(0,2)

---ВЫХОДЫ НЕЙРОНОВ 3-го СЛОЯ-----

Синдромы 3-го уровня:

Синдром3_1=Сигмоида3(0,2*Синдром2_1-0,2*Синдром2_2+0,3*Синдром2_3+0,01483843)

Синдром3_2=Сигмоида3(0,3*Синдром2_1-0,2*Синдром2_2+0,3*Синдром2_3+0,03120456)

---НОРМАЛИЗОВАННЫЙ ВЫХОД НЕЙРОСЕТИ-----

Конечные синдромы:

Z=-0,9*Синдром3_1-0,7*Синдром3_2+0,1

---ВЫХОД НЕЙРОСЕТИ-----

Постобработка конечных синдромов:

Z=((Z*1)+1)/2)

Задание на лабораторную работу

В работе требуется на основе предлагаемой статистики создать ИНС, провести обучение и тестирование ИНС.

1. Запустить NeuroPro.
2. Создать новый проект (Файл -> Создать).
3. Выбрать файл со статистикой задачи (Открыть файл данных); в случае удачного открытия файла статистики он отображается на экране в виде таблицы.
4. Переключиться на окно проекта (окно с таблицей не закрывать).
5. Создать новую нейросеть(“Новая сеть”-> “Создание нейронной сети”).
6. Настроить параметры ИНС («Входы и выходы», «Структура сети»).
7. Сохранить проект в файл (Файл -> Сохранить).

8. Произвести поиск конфликтных примеров в обучающей выборке (Нейросеть -> Анализ обучающего множества); в случае обнаружения конфликтных примеров удалить их, затем повторить проверку;

9. Провести обучение ИНС (Нейросеть -> Обучение);

10. Провести тестирование ИНС (Нейросеть -> Тестирование)