

УДК 519.95

АВТОМАТИЗИРОВАННАЯ ТЕХНОЛОГИЯ И ПРОГРАММНО-ТЕХНОЛОГИЧЕСКИЙ КОМПЛЕКС ДЛЯ ПОСТРОЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ С ИНТЕЛЛЕКТУАЛЬНЫМ ЯДРОМ, ОСНОВАННОМ НА НЕЙРОСЕТЕВЫХ МОДЕЛЯХ, ПОДДЕРЖКОЙ РАСПРЕДЕЛЕННОГО ВВОДА ДАННЫХ И ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ¹

© А.А. Арзамасцев, Н.А. Зенкова, О.В. Крючин, Д.О. Квашенкин, А.В. Неудахин

Ключевые слова: автоматизированная технология; экспертные системы; искусственная нейронная сеть; интеллектуальное ядро; параллельные вычисления; распределенный ввод данных; механизмы удаленного доступа. В результате выполнения проекта разработан программно-технологический комплекс, позволяющий создавать экспертные системы для различных сфер деятельности. Экспертные системы этого типа базируются на распределенном вводе исходных данных с помощью системы удаленного доступа, интеллектуальном ядре, на основе искусственных нейронных сетей, при его поэтапном обучении с использованием параллельных алгоритмов. Программный комплекс прошел апробацию при разработке экспертных систем в следующих областях: промышленный объект (экспертная система для технологического процесса выработки биомассы на основе отходов производства этанола); медицинский объект (экспертная система для прогнозирования исхода хирургического лечения осложненного колоректального рака). С его использованием получены также модели для экологических объектов (прогнозирование плотности популяции креветки в открытой системе и температуры воздуха в г. Тамбове); экономических объектов (прогнозирование котировок валют); социальных объектов (модель профессиональной предрасположенности школьников старших классов). Во всех случаях достигнуто высокое качество соответствия эмпирических и расчетных значений за счет высокой адаптационной способности ядра экспертной системы и процедуры поэтапного обучения. В ходе работы выявлены: универсализм предложенных механизмов формирования и адаптации экспертной системы к эмпирическим данным из различных предметных областей, возможность практического использования данных приемов и методов. Разработанный комплекс программ может иметь коммерческое использование.

ВВЕДЕНИЕ

Необходимость в экспертных системах (ЭС) как в интеллектуальных помощниках, способных решать трудноформализуемые задачи, существует во многих сферах человеческой деятельности: промышленности, научных исследованиях, социальной и экономической областях.

Вместе с тем создание и эксплуатация таких систем характеризуется в настоящее время рядом трудностей и ограничений: высокой трудоемкостью процессов построения баз знаний ЭС, связанной с формализацией знаний экспертов и их машинным представлением, а в некоторых случаях невозможностью такой формализации вследствие большого количества невербальных данных, влияющих на оценку эксперта; необходимостью привлечения дополнительного круга специали-

стов – инженеров по знаниям и разработке специализированного программного обеспечения или адаптации существующих ЭС под конкретные условия рассматриваемой предметной области; преобладанием локального подхода к организации взаимодействия с ЭС, тогда как в настоящее время все большее распространение и доступность получают сетевые информационные системы, например, веб-ориентированные.

Таким образом, становится актуальной задача разработки новой информационной технологии, позволяющей устранить вышеуказанные трудности и ограничения путем автоматизированного формирования базы знаний для конкретной предметной области на основе распределенного сетевого ввода данных.

Решение данной задачи возможно за счет использования в качестве системы представления знаний в ЭС моделей искусственных нейронных сетей (ИНС-моделей). Применение таких моделей позволит обеспечить автоматизированное формирование развивающейся (по мере поступления информации от источников) системы представления знаний, отличающейся высокой степенью адаптируемости к эмпирическим данным и существенной гибкостью за счет алгоритмов выбора начальной структуры ИНС и ее модификации.

Поскольку формирование развивающейся системы представления знаний ЭС предполагает ее совершенствование на основе постоянно поступающих новых данных от распределенных терминалов, в качестве одного

¹ В статье представлены результаты исследований, выполненных по государственному контракту №8894р/14330 от 11.04.2011 г. «Разработка автоматизированной технологии и программно-технологического комплекса для построения экспертных систем для различных сфер деятельности, включающей оригинальные способы и программные механизмы разработки экспертных систем, поддерживающих распределенный ввод данных, параллельные вычисления и имеющих интеллектуальное ядро, основанное на нейросетевых моделях». Фонд содействия развитию малых форм предприятий в научно-технической сфере. Программа СТАРТ. 2011–2012 гг.

из главных условий реализации данного класса ЭС выступает разработка системы сетевого доступа, позволяющей обеспечить передачу информации, а также управление ЭС посредством веб-интерфейса.

Преимуществами предлагаемых способов решения задачи по разработке информационной технологии являются: отсутствие пространственных ограничений по взаимному расположению источников данных, самой ЭС и ее пользователей; независимость процедуры формирования базы знаний от области использования ЭС; автоматизированная генерация знаний.

В настоящее время искусственные нейронные сети нашли широкое применение в системах искусственного интеллекта, в т. ч. и интеллектуальных экспертных системах.

Главными недостатками ИНС как универсального аппарата построения моделей для объектов, заданных эмпирическими данными, являются: значительная сложность алгоритмов выбора структуры сети и существенная размерность вектора переменных целевой функции, минимизируемой при ее обучении. Эти недостатки приводят к значительным временным затратам в процессе разработки ИНС-модели, адекватной реальному объекту.

Одним из наиболее перспективных методов решения данной проблемы является разработка параллельных алгоритмов построения ИНС-модели при использовании кластерных вычислительных систем. Поскольку задача построения ИНС-модели предполагает разработку различных параллельных алгоритмов (минимизации целевой функции, реализации соответствующих методов нелинейного программирования, модификации структуры сети и т. д.) и их взаимодействия, следует говорить о разработке соответствующей автоматизированной информационной технологии и программно-технологического комплекса.

Рассмотрению экспертных систем как классу систем искусственного интеллекта посвящены многочисленные работы российских и зарубежных ученых [1–21].

На основе рассмотренных научных работ [14, 22–35] и существующих примеров реализации область применения ЭС наиболее широко можно классифицировать по типу решаемых задач: диагностика, прогнозирование, планирование, проектирование, интерпретация, контроль и управление, обучение.

В настоящее время наиболее широкое распространение получили т. н. традиционные, или типичные ЭС. На основе [14, 22–35] был проведен анализ структуры и режимов функционирования типичной ЭС. В общем виде типичную ЭС можно представить следующим образом.

Типичная ЭС состоит из следующих *основных компонентов*: решателя (интерпретатора); рабочей памяти (РП), называемой также базой данных (БД); базы знаний (БЗ); компонентов приобретения знаний; объяснительного компонента; диалогового компонента.

Экспертная система может работать в двух режимах: режиме приобретения знаний и в режиме решения задачи (называемом также режимом консультации или режимом использования ЭС).

Разработка ЭС имеет существенные отличия от разработки обычного программного продукта. Опыт создания ЭС, освещенный в научных работах [15,

36–46], показал, что использование при их разработке методологии, принятой в традиционном программировании, либо чрезмерно затягивает процесс создания ЭС, либо вообще приводит к отрицательному результату. Использовать ЭС следует только тогда, когда разработка ЭС возможна, оправдана и методы инженерии знаний соответствуют решаемой задаче. Можно выделить работу [30] как логическое обобщение особенностей задач, решаемых с помощью ЭС.

Согласно работам российских и зарубежных ученых [30, 47–48] по созданию ЭС, сложилась определенная технология их разработки, включающая шесть этапов: идентификацию, концептуализацию, формализацию, выполнение, тестирование, опытную эксплуатацию.

Проблеме программной реализации ЭС посвящены многие труды российских и зарубежных ученых [1–21]. Необходимо отметить, что использование аппарата искусственных нейронных сетей для построения ИНС-моделей различных объектов признано успешным в работах [49–54], а в работах [55–57] показана принципиальная применимость ИНС-моделей в разработке экспертных систем.

В работах [48, 58] указывается на принципиальную возможность использования удаленного доступа при организации ЭС.

Таким образом, в ходе краткого обзора исследований и литературных источников по проблемам ЭС выявлено, что в настоящее время не существует информационных технологий и программно-технологических комплексов, сочетающих в себе преимущества удаленного доступа, интеллектуального ядра на основе ИНС-моделей и параллельных алгоритмов их обучения при проектировании экспертных систем широкого назначения.

По этой причине следует считать актуальной тему НИОКР «Разработка автоматизированной технологии и программно-технологического комплекса для построения экспертных систем для различных сфер деятельности, включающей оригинальные способы и программные механизмы разработки экспертных систем, поддерживающих распределенный ввод данных, параллельные вычисления и имеющих интеллектуальное ядро, основанное на нейросетевых моделях».

1. РАЗРАБОТКА ОБЩЕЙ КОНЦЕПЦИИ ТЕХНОЛОГИИ И АВТОМАТИЗИРОВАННОГО ПРОГРАММНО-ТЕХНОЛОГИЧЕСКОГО КОМПЛЕКСА

1.1. Теоретические основы

Для объединения функциональных преимуществ нейросетевых и веб-ориентированных ЭС разработана концептуальная модель веб-ориентированной нейросетевой экспертной системы (рис. 1.1). Основными отличительными компонентами такой модели от традиционной являются: веб-интерфейс, позволяющий осуществлять распределенный ввод данных; база данных, используемая для долговременного хранения технической информации; развивающаяся система представления знаний экспертов, основанная на нейросетевой модели. Согласно разработанной модели, решателем

является построенная и обученная ИНС, генерирующая решение задачи на основе эмпирических данных, хранящихся в БД.

Для разработки информационной технологии построения ЭС, соответствующих предлагаемой концептуальной модели, выполнена математическая формализация задач накопления информации и построения на ее основе развивающихся систем представления знаний.

Основные принципы формализации. Пусть имеется распределенная система получения исходной информации, включающая N терминалов, каждый из которых предназначен для ввода n независимых параметров, имеющих m уровней каждый. Работа терминалов организована таким образом, что каждый из них работает лишь часть времени, так, что интервалы их простоя являются случайными величинами, заданными плотностями распределений $p_1(\tau), p_2(\tau), \dots, p_N(\tau)$. Время ввода информации не зависит от номера терминала, но линейно зависит от n , т. е. $\tau_{\text{input}} = k_1 n$, где k_1 – некоторый коэффициент пропорциональности. Будем считать, что очередная доля информации добавляется к БД в виде одной записи, когда на i -м терминале ($i = 1, \dots, N$) каждому независимому параметру x_j ($j = 1, \dots, n$) присвоен соответствующий уровень l_b ($b = 1, \dots, m$). Также учитывается, что передача информации с любого терминала в БД осуществляется мгновенно, т. к. это время на несколько порядков меньше времени, затрачиваемого пользователем на ввод информации.

Тогда среднее время одного цикла работы i -го терминала (ввод + ожидание следующего ввода информации), соответствующее добавлению в базу данных одной записи, может быть вычислено как:

$$\bar{\tau}_i = k_1 n + \int_0^{t_i^{\max}} \tau p_i(\tau) d\tau.$$

Поэтому за период времени T (например, $T = 24$ ч) на i -м терминале могут быть осуществлены r_i циклов ввода, соответствующие добавлению в базу данных r_i записей:

$$r_i = \frac{T}{\bar{\tau}_i} = \frac{T}{k_1 n + \int_0^{t_i^{\max}} \tau p_i(\tau) d\tau}.$$

Необходимо отметить, что r_i является целым числом, полученным в результате округления в большую сторону полученного из уравнения значения. Соответственно, общее число записей информации в БД, полученное со всех терминалов за период T , составит:

$$R = \sum_{i=1}^N r_i = T \sum_{i=1}^N \frac{1}{k_1 n + \int_0^{t_i^{\max}} \tau p_i(\tau) d\tau}. \quad (1.1)$$

Выполним оценку периода первичного накопления данных в БД, предшествующего первому обучению ИНС-модели.

В соответствии с теоремами А.Н. Колмогорова [59–60] о представимости функций нескольких переменных с помощью суперпозиций и сумм функций одного переменного можно утверждать, что каждая непрерывная функция n переменных (в ЭС n соответствует количеству вводимых независимых параметров), заданная на единичном кубе n -мерного пространства, представима

в виде: $f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} h_q \left[\sum_{p=1}^n \varphi_q^p(x_p) \right]$, где

функции $h_q(u)$ непрерывны, а функции $\varphi_q^p(x_p)$, кроме того, еще и стандартны, т. е. не зависят от выбора функции f .

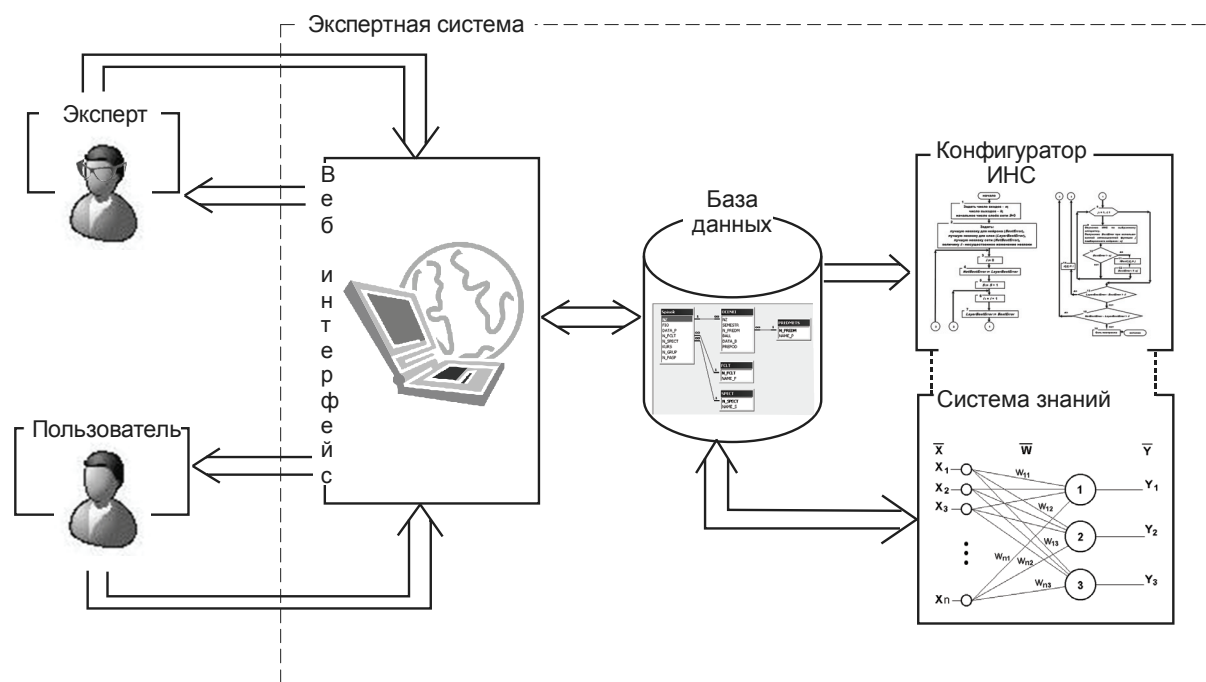


Рис. 1.1. Концептуальная модель веб-ориентированной нейросетевой ЭС

Фактически это означает, что «минимальная» структура ИНС-модели, позволяющая аппроксимировать функцию n переменных, должна иметь следующий вид: входной слой, состоящий из n (n соответствует переменным функции) нейронов; первый скрытый слой, включающий $(2n + 1) \cdot n$ функциональных нейронов; второй скрытый слой, состоящий из $(2n + 1)$ функциональных нейронов, и выходной, суммирующий нейроны (рис. 1.2). Указанная нейронная сеть не является полносвязанной и имеет $(2n + 1) \cdot n + (2n + 1) \cdot n + 2n + 1 = (2n + 1)^2$ связей (степеней свободы при обучении ИНС-модели).

Необходимо заметить, что теоремы А.Н. Колмогорова не несут никакой информации о типе нелинейности функций φ и h . Так, в системе знаний экспертной системы, базируемой на ИНС-модели, используется активационная функция $f(x) = \frac{1}{1 + e^{-\alpha x}}$ и полносвязанная сеть – многослойный персептрон. Для этого случая число степеней свободы ИНС-модели будет $(2n + 1) \cdot (3n^2 + n + 1)$. Учитывая, что число записей в БД должно быть не меньше, чем число степеней свободы, получим оценку для первичного периода накопления данных:

$$t_1 \geq K \cdot \left(\sum_{i=1}^N \frac{1}{k_1 n + \int_0^{t_i^{\max}} \tau p_i(\tau) d\tau} \right)^{-1}, \quad (1.2)$$

где K – параметр, характеризующий число степеней свободы: $K = (2n + 1)^2$ для ИНС-модели, построенной в соответствии с теоремой А.Н. Колмогорова, и $K = (2n + 1) \times (3n^2 + n + 1)$ для полносвязанной ИНС-модели.

Процесс обучения модели знаний ЭС заключается в минимизации функционала вида:

$$E[w(n)] = \|\Delta\| = \|Y^{tabl} - Y^{net}(w)\| \rightarrow \min_w. \quad (1.3)$$

Вектор w определяется структурой сети, которая, в свою очередь, связана с числом ее входов n (n соответствует числу нейронов входного слоя ИНС-модели) так, что имеется однозначное соответствие n и w . Поэтому вектор w^* , зависящий от структуры сети и минимизирующий функционал (1.3), соответствует оптимальной ИНС-модели и может быть определен как:

$$w^* = \arg \min_{w \in \Omega} F(w), \quad (1.4)$$

где Ω – множество векторов весовых коэффициентов нейронов.

Если минимизацию функционала (1.3) осуществлять двумя методами: градиентным, так что последующие значения вектора w вычисляются по формуле $w^{t+1} = w^t - h(t) \text{grad}(E(w^t))$, и методом сканирования, то для указанных методов оценки времени обучения

могут быть получены следующим образом: $\Theta_{grad} = k_2 n R$ и $\Theta_{scan} = k_3 R \alpha^N$. Здесь k_2 и k_3 – коэффициенты, зависящие от технических характеристик используемого оборудования, α – число разбиений диапазона изменения переменной.

Целью обучения ИНС-модели на основе эмпирических данных является нахождение подходящей структуры и коэффициентов ИНС, минимизирующих невязку (1.3). Указанная структура обобщает в себе знания, полученные на основе данных, поступивших с терминалов в БД.

Данные, используемые для обучения ИНС-модели, могут различаться как в количественном отношении (число записей в БД на момент начала обучения), так и в качественном (дисперсия, наличие ошибок и т. д.).

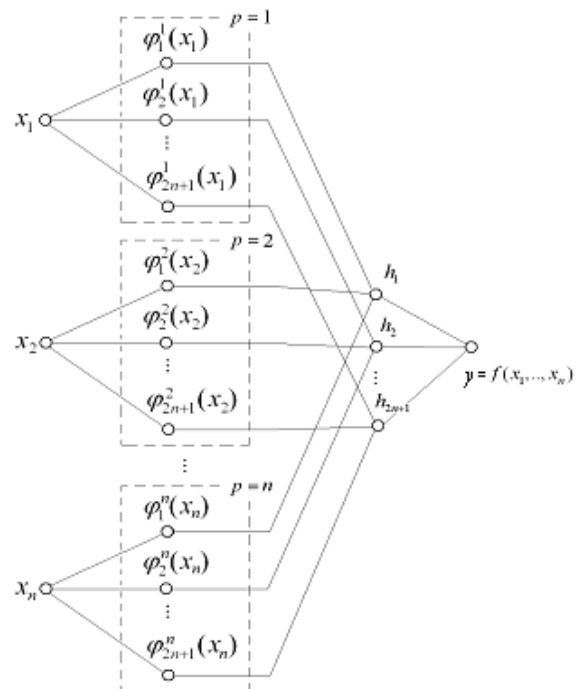


Рис. 1.2. Представление функции n переменных в виде ИНС-модели

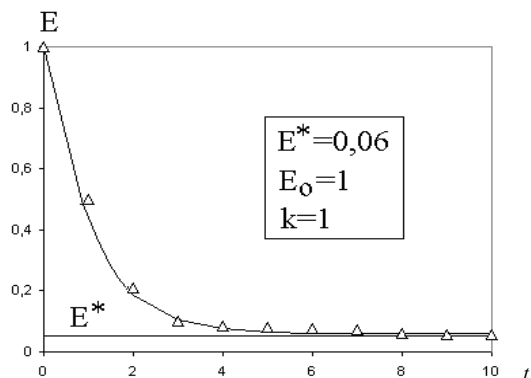


Рис. 1.3. Корреляция эмпирических данных по среднеквадратичной погрешности обучения сети – E (треугольники) от времени обучения с результатами расчета по уравнению (1.5). На графике показаны приведенные безразмерные значения

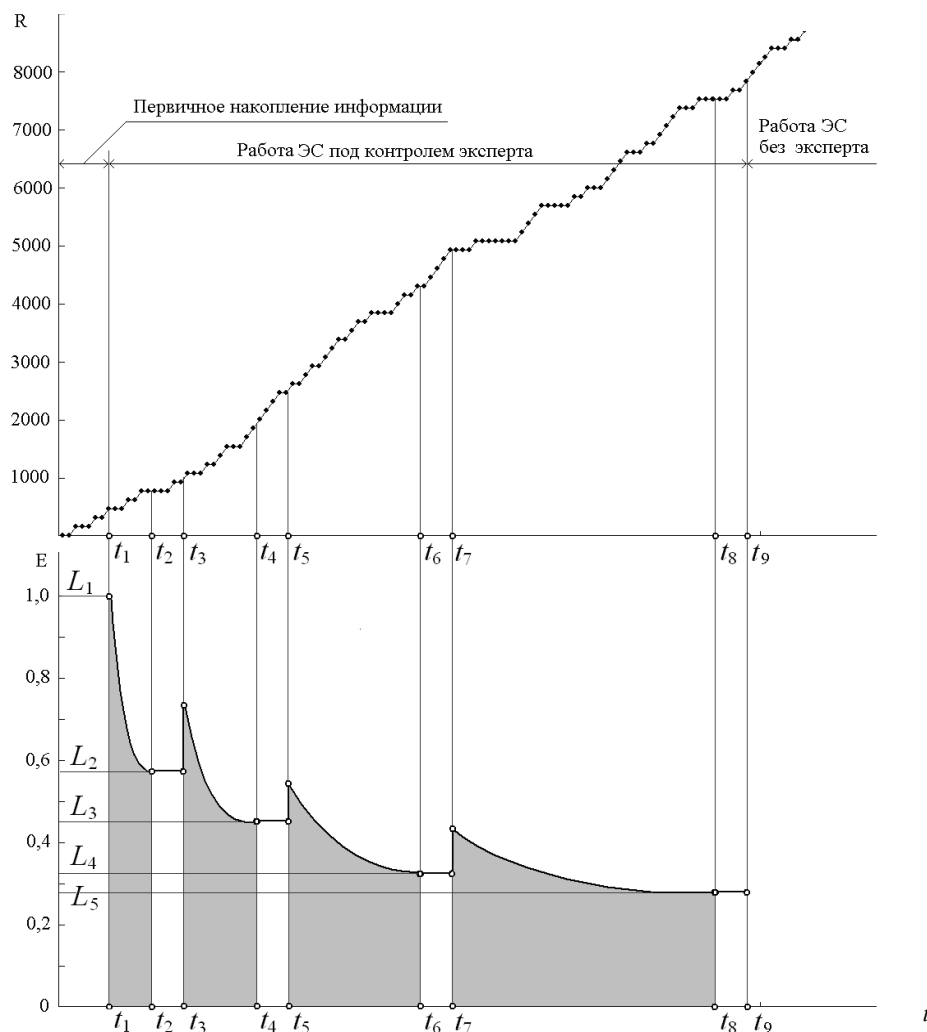


Рис. 1.4. Зависимость накопления первичной информации от безразмерного времени (верхняя часть рисунка): R – число записей в БД, полученных с терминалов; процесс обучения ИНС-модели (нижняя часть рисунка): E – приведенная среднеквадратическая ошибка ИНС-модели, L – уровни ее обученности

Поэтому разработка аналитической модели обучения для различных объектов не позволяет использовать теоретические подходы. В качестве эмпирической модели обучения предлагается использовать следующее дифференциальное уравнение:

$$\frac{dE(t)}{dt} = k_4 [E^* - E(t)], \quad (1.5)$$

с начальным условием: $E(0) = E_0$,

где $E(t)$, E_0 и E^* – значения приведенной среднеквадратичной погрешности, ее начального значения и уровня, на котором она будет зафиксирована в конце цикла обучения ИНС-модели; k_4 – параметр удельной скорости обучения, зависящий от n , объема обучающей выборки и используемого метода минимизации невязки (1.3); t – безразмерное время процесса обучения, полученное отнесением реального времени к времени первого цикла обучения $t \in [t_1, t_2]$ (рис. 1.3, 1.4). Уравнение (1.5) отражает феноменологию процесса обучения: в начальный момент времени невязка равна E_0 и стремится к E^* в конце цикла обучения.

Разделяя переменные в уравнении (1.4) и интегрируя, получим:

$$\int \frac{dE(t)}{k_4 (E^* - E(t))} = \int dt, \quad \text{или} \quad -\frac{1}{k_4} \ln[k_4 (E^* - E(t))] = t + c. \quad (1.6)$$

Определив постоянную интегрирования из начальных условий (1.5), получим решение в виде:

$$E(t) = E^* (1 - e^{-k_4 t}) + E_0 e^{-k_4 t}. \quad (1.7)$$

Предложенные аналитические модели на практике позволяют решить следующие задачи: оценить первичный период накопления информации, необходимой для построения ИНС-модели исследуемого объекта; определить начальную структуру ИНС-модели на основе входных характеристик объекта исследования; оценить количество циклов обучения ИНС-модели в присутствии

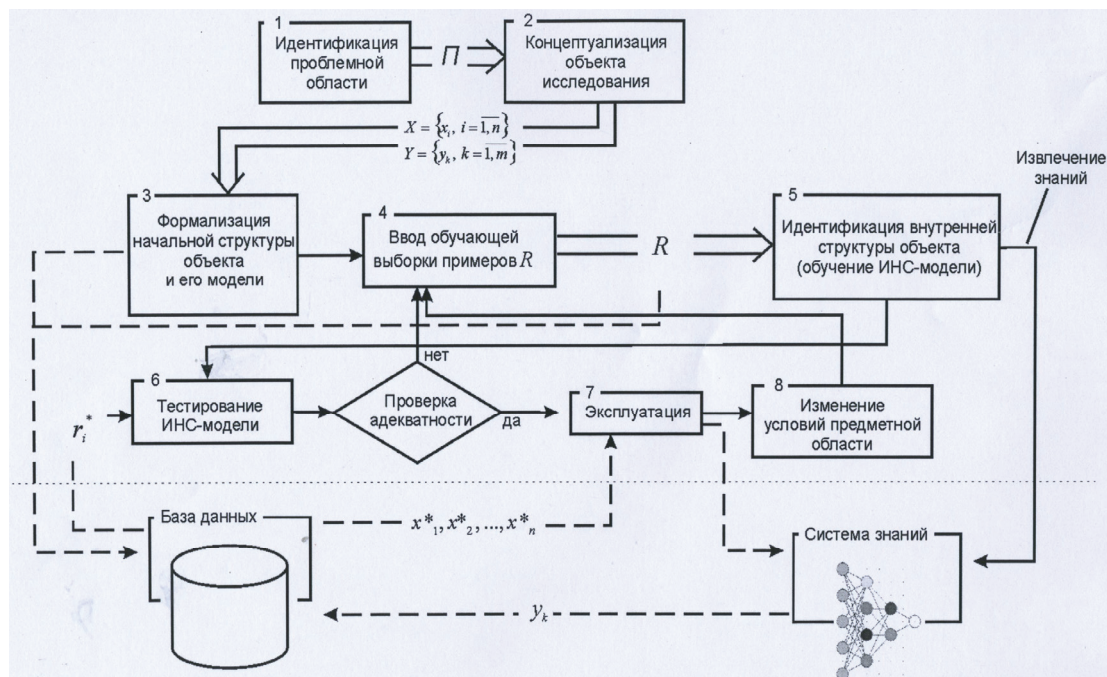


Рис. 1.5. Функциональная модель жизненного цикла ЭС на основе нейросетевой модели и с распределенным вводом данных

эксперта. Эти результаты позволили разработать функциональную модель жизненного цикла ЭС на основе ИНС и поддерживающей распределенный сетевой ввод данных (рис. 1.5). Входом ЭС является запись из совокупности эмпирических данных предметной области, а выходом – номер(а) классов, к которым она принадлежит.

Автоматизированная информационная технология построения ЭС на основе нейросетевой модели и с распределенным вводом данных, соответствующая функциональной модели ЭС (рис. 1.5), заключается в последовательном выполнении следующих шагов:

- изучение объекта, для которого разрабатывается ЭС; выделение векторов входных характеристик и возможных выходных состояний данного объекта;
- накопление первичной информации об объекте в БД посредством распределенной сетевой системы сбора данных в течение периода, определяемого аналитической моделью;
- формирование первичной структуры ИНС-модели;
- планирование поэтапного процесса обучения развивающейся ИНС-модели по предложенному уравнению (1.7); обучение ИНС-модели согласно предложенным аналитическим моделям;
- совершенствование ИНС-модели на основе вновь поступающих эмпирических данных, под контролем эксперта, который определяет законченность формирования системы представления знаний;
- эксплуатация ЭС (без эксперта).

На основе сравнения разработанной технологии с существующими аналогами выделены ее функциональные преимущества, заключающиеся в следующем: универсальность – применительно к различным предметным областям (без внесения конструктивных изменений в программное обеспечение); многопользователь-

ский доступ к одному объекту исследования; автоматизация процесса проектирования базы знаний за счет конструктивных алгоритмов построения ИНС-модели; поддержка сетевых технологий взаимодействия.

Выявлены условия, при которых может быть осуществлено применение разработанной технологии, такие как: возможность формализовать начальную структуру объекта и закодировать примеры его диагностики в числовом виде; наличие эксперта, обладающего высоким профессиональным уровнем (выше уровня рядового специалиста), способного адекватно оценивать параметрические данные объекта; достаточная квалификация пользователей, имеющих опыт использования сервисов, предоставляемых консорциумом WWW.

1.2. Структура программно-технологического комплекса для автоматизированного построения экспертных систем при использовании удаленного доступа к источникам информации и параллельных алгоритмов обучения интеллектуального ядра

На основе обзора существующих разработок экспертных систем (см. введение) установлено, что большинство из них обладают решателем со строго организованной структурой, которая изначально закладывается в него инженером по знаниям и впоследствии практически не изменяется. Поскольку условия предметных областей, в которых применяются экспертные системы, часто претерпевают изменения, и, соответственно, изменяются приоритеты факторов, влияющих на решение задач, нами было выбрано направление исследований, включающих разработку программных механизмов, отвечающих за решающие способности экспертной системы, которые могли бы поэтапно совершенствоваться в процессе работы экспертной системы. Данные

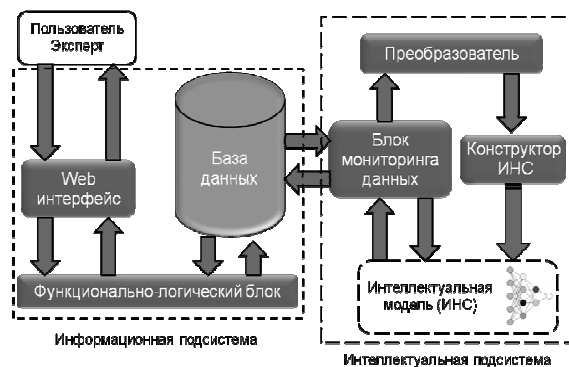


Рис. 1.6. Общая структура организации программного комплекса

механизмы улучшают адаптируемость экспертной системы к условиям различных предметных областей. Также в данном проекте предлагается использовать удаленный доступ для ввода данных в систему и для управления процессами разработки, совершенствования и эксплуатации экспертных систем. Общая структура организации комплекса представлена на рис. 1.6.

В ходе предварительных вычислительных экспериментов было показано, что наиболее «узким» местом в работе всей технологии и программно-технологического комплекса является обучение интеллектуального ядра системы – ИНС-модели. В различных ситуациях время обучения такой модели изменялось от одного часа до двух недель, что не могло быть признанным удовлетворительным. По этой причине было принято решение об использовании в программно-технологическом комплексе параллельных методов обучения ИНС-модели с отсылкой структуры сети, ее настроек и эмпирических данных на удаленный вычислительный кластер.

Поэтому система сконфигурирована таким образом, что программный комплекс состоит из 6 подсистем (рис. 1.7): *Manager* – подсистема управления; *ANNBuilder* – подсистема построения ИНС-модели; *ANNExecutor* – подсистема эксплуатации ИНС-модели; *DB* – база данных; *UserMediator* – подсистема взаимодействия с пользователями; *Logger* – подсистема сбора логов.

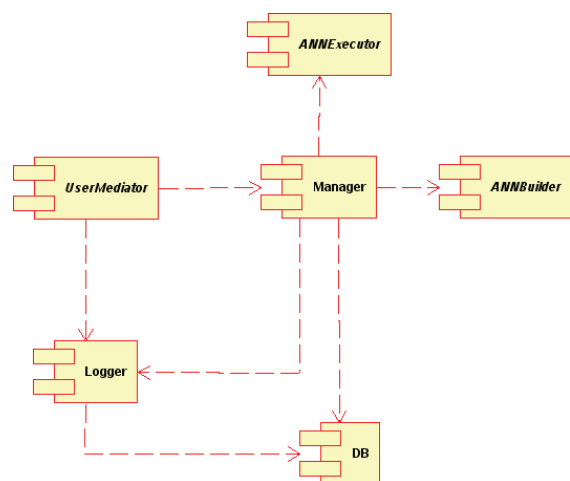


Рис. 1.7. Диаграмма подсистем комплекса

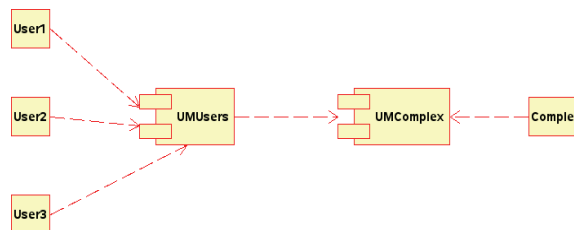


Рис. 1.8. Диаграмма компонентов *UserMediator*

***UserMediator*.** Подсистема *UserMediator* представляет собой совокупность двух компонентов, являющихся серверами (рис. 1.8), первый из которых (*UMUsers*) предназначен для подключения пользователей из локальной или глобальной сети. В качестве сервера используется *Web*-сервер *Apache*, а в качестве протокола – *HTTP*. Второй сервер (*UMComplex*) предназначен для подключения комплекса, а в качестве протокола используется *EXSP* (формат сообщений текстовый, сообщение состоит из набора пар ключ-значение, разделяемых точкой с запятой).

***UMUsers*.** Данный компонент предназначен для взаимодействия с пользователями. Поскольку для взаимодействия используется протокол *HTTP*, то пользователям для работы с комплексом необходим только *Web*-браузер (*Opera*, *Mozilla Firefox*, *Google Chrome*, *Safari* или *Konqueror*). Интерфейс реализован на языке *PHP*, для создания экранов используются шаблоны *Smarty*.

Для взаимодействия с каждым пользователем создается отдельный поток, таким образом, множество пользователей может работать с Комплексом одновременно. При подключении нового пользователя *UMUsers* устанавливает соединение (новое) с *UMComplex* и передает ему запрос пользователя. При этом запрос конвертируется из формата *HTTP* в *EXSP*. Получив ответ от *UMComplex*, *UMUsers* преобразует его из формата *EXSP* в *HTTP*.

***UMComplex*.** Данный компонент предназначен для взаимодействия между компонентом *UMUsers* и остальными компонентами Комплекса. Подключаясь к нему, *UMUsers* и *Manager* посылают свой идентификатор. При пересылке запроса, поступившего от *UMUsers*, *UMComplex* добавляет в него идентификатор пользователя. Это необходимо в связи с тем, что к *UMComplex* имеется множество подключений *UMUsers*, но только одно подключение *Manager* (Комплекса).

***ANNBuilder*.** Данная подсистема предназначена для построения ИНС-модели. Она размещается на кластерной системе и состоит из двух компонентов – *UNS* и *UNSS*. Первый из них представляет собой нейросетевой симулятор и располагается на вычислительных узлах кластерной системы, а второй – сервер симуляторов. Оба компонента реализованы на языке *C++*.

***UNS*.** Нейросетевой симулятор, размещаемый на вычислительных узлах кластерной системы, представляет собой программу с распределенной памятью и запускается через систему запуска вычислительного кластера (в случае вычислительного кластера ТГУ им. Г.Р. Державина это *PBS*). В зависимости от количества имеющихся вычислительных узлов и требуемых процессоров может одновременно выполняться один или

несколько экземпляров *UNS* (имеется в виду одна или несколько совокупностей программ).

UNSS. Сервер симуляторов, размещаемый на мастер-узле (сервере) кластерной системы, предназначен для осуществления взаимодействия между *UNS* и *Manager*. При необходимости построения ИНС-модели *UNSS* формирует задание (*PBS* или иное в зависимости от архитектуры вычислительного кластера) и запускает *UNS*. Поскольку количество одновременно работающих *UNS* ограничено аппаратными ресурсами (количество вычислительных узлов) кластерной системы, то *UNSS* ставит задачи в очередь и при освобождении ресурсов запускает их.

ANNExecuter. Подсистема *ANNExecuter* предназначена для выполнения экспертиз (вычисления выходных значений ИНС-модели от вектора входных). Установивший с ней соединение *Manager* передает структуру ИНС-модели и вектор входных данных. Подсистема состоит из одного компонента, который реализован на языке C++.

DB. Данная подсистема состоит из двух компонентов – *DBMediator*, который предназначен для осуществления взаимодействия между базой данных и *Manager* и СУБД. Система предназначена для хранения данных (виды хранимой информации приведены ниже).

DMMediator. Данный компонент предназначен для преобразования запросов из формата комплекса *EXSP* в *SQL*. Для хранения данных используется СУБД *Postgree*, поэтому для взаимодействия с базой данных используется библиотека *pgx*. Компонент написан на языке C++.

Хранимая информация. В базе данных хранится следующая информация: параметры моделируемых объектов (отвечающие за структуру объекта); обучающая выборка; структуры ИНС; пользователи Комплекса; экспертизы (данные эксплуатации модели).

Logger. Данная подсистема, состоящая из одного компонента, предназначена для логизации обмена информацией. *Manager* дублирует все посылаемые и получаемые сообщения в данную подсистему. *Logger* все получаемые сообщения посылает в *DB* для сохранения в базу данных, откуда они могут быть прочитаны. Компонент реализован на языке C++.

Manager. Данная подсистема представляет собой центральный компонент Комплекса и компоненты передачи данных, которые необходимы в случае размещения прочих подсистем на узлах, к которым нет прямого доступа. Все компоненты реализованы на языке C++.

После запуска *Manager* устанавливает соединение с *UserMediator*, *DB*, *ANNExecuter* и *ANNBuilder*. В зависимости от запросов, поступающих с *UserMediator*, он строит или использует ИНС-модель.

1.3. Организация параллельных вычислений при обучении интеллектуального ядра программно-технологического комплекса

Параллельная программа имеет особый принцип работы, поскольку, в отличие от последовательной, где одновременно выполняется только одно действие, в параллельных программах одновременно выполняются

несколько операций. Поскольку параллельная программа выполняется на многопроцессорной машине, то необходимо задействовать все процессоры, т. е. дать им на выполнение какую-то часть задачи, которая может выполняться самостоятельно (не задействуя при этом другие части задачи).

Обучение искусственной нейронной сети (ИНС) сводится к минимизации целевой функции:

$$\varepsilon = \|d - y\| = \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{p-1} (d_{i,j} - y_{i,j})^2} = \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{p-1} (d_{i,j} - F(x_i, w))_j^2}. \quad (1.8)$$

Здесь d , y представляют собой выходные значения самого объекта (таблица эмпирических данных) и ИНС при одинаковых и заданных входах; x – входные данные; N – число строк данных в обучающей выборке; p – размерность вектора выходных координат объекта; w – вектор коэффициентов синаптических (межнейронных) связей; i – номер текущей строки обучающей выборки; j – номер текущего выхода. Минимизация невязки (1.8) обычно заключается в подборе вектора весовых коэффициентов w при помощи какого-либо метода. Таким образом, представляется очевидным выделить два уровня, на которых можно производить распараллеливание – вычисление значения функции (1.8) и подбор вектора w .

Распараллеливание на уровне вычисления значения целевой функции. При обучении сети вычисление выходных сигналов требуется для их сравнения с выходными данными из обучающей выборки, которое, в свою очередь, необходимо для вычисления погрешности (1.8). Поскольку обучающая выборка, как правило, состоит из нескольких сотен (или даже тысяч) строк данных, а количество процессоров не превышает нескольких десятков, можно равномерно распределить строки обучающей выборки по процессорам с тем, чтобы после передачи с управляющего узла структуры сети все процессоры одновременно вычисляли выходные сигналы, сравнивали их со значениями из обучающей выборки и передавали обратно на ведущий процессор лишь значение погрешности. Процессор управляющего узла суммирует эти погрешности и получает, таким образом, результат – значение целевой функции. Таким образом, при необходимости вычислить эту функцию управляющий узел рассылает на все узлы структуру нейронной сети и получает от них погрешность, полученную на той части обучающей выборки, которая была передана на этот вычислительный узел.

Таким образом, формулу (1.8) можно представить в виде:

$$\varepsilon = \sum_{k=0}^{n-1} \varepsilon_k = \sum_{k=0}^{n-1} \sum_{i=k*\hat{N}}^{(k+1)*\hat{N}} \sum_{j=0}^{p-1} \sqrt{(d_{i,j} - F(x_i, w))_j^2}, \quad (1.9)$$

где ε_k – погрешность, вычисленная на k -м процессоре;

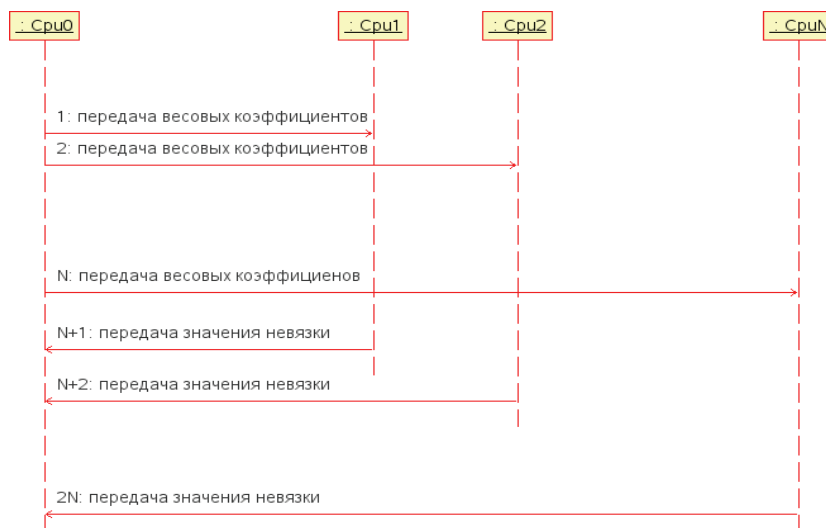


Рис. 1.9. Диаграмма последовательности вычисления значения невязки

n – количество процессоров; $\hat{N} = \frac{N}{n}$ – количество строк обучающей выборки, передаваемой на каждый процессор.

В случае, когда в процессе обучения ИНС структура не меняется, т. е. происходит только подбор весовых коэффициентов, то для получения значения невязки на вычислительные узлы передаются только значения весовых коэффициентов (рис. 1.9), а обучающая выборка и структура сети передаются один раз до начала обучения.

Распараллеливание на уровне изменения весовых коэффициентов при использовании алгоритма полного сканирования. Одним из самых простых является алгоритм полного сканирования. Как известно, алгоритм полного сканирования работает следующим образом: поскольку весовые коэффициенты обычно задаются в ограниченном интервале $[l_0; l_1]$ (как правило, $l_0 = -1$, $l_1 = 1$), то вначале происходит инициализация всех весовых коэффициентов минимальным значением, а на следующих итерациях происходит последовательное изменение.

$$w = \{w_1, w_2, w_3, \dots, w_L\} \quad w_k = w_{k0} + is, \quad (1.10)$$

$$k = 1, \dots, L, \quad i = 0, 1, 2, \dots, \frac{l_1 - l_0}{s} + 1,$$

где s – заданный шаг. То есть происходит полный перебор всех возможных комбинаций весовых коэффициентов.

Таким образом, после вычисления значения целевой функции при существующей конфигурации значение последнего весового коэффициента увеличивается на величину шага s , и снова происходит вычисление значения целевой функции. После того как значение последнего весового коэффициента достигнет верхнего предела l_1 , предпоследний коэффициент увеличивается на величину шага, а последний устанавливается минимальным пределом l_0 и т. д. После того как значе-

ния всех весовых коэффициентов будут соответствовать верхнему пределу, перебор заканчивается. Таким образом, происходит полный перебор всех возможных комбинаций значений весовых коэффициентов.

Этот способ, с одной стороны, является довольно надежным, а с другой – очень медленным. Количество итераций можно вычислить по формуле:

$$I = \left(\frac{l_1 - l_0}{s} + 1 \right)^L, \quad (1.11)$$

где L – количество весовых коэффициентов, s – величина приращения весового коэффициента.

Параллельное выполнение этого метода заключается в том, что для каждого процессора значения $\hat{L}$ первых весовых коэффициентов являются фиксированными (на первом процессоре $w_0 = l_0, w_1 = l_0, \dots, w_{\hat{L}-1} = l_0$, на втором $w_0 = l_0, w_1 = l_0 + s, w_{\hat{L}-1} = l_0 + s$ и т. д.), а остальные подбираются обычным способом (рис. 1.10). После того как все процессоры переберут возможные варианты и каждый из них выберет оптимальную структуру и вычислит погрешность при этой структуре, эти результаты передаются ведущему процессору, который сравнивает значения невязки и выбирает наимень-

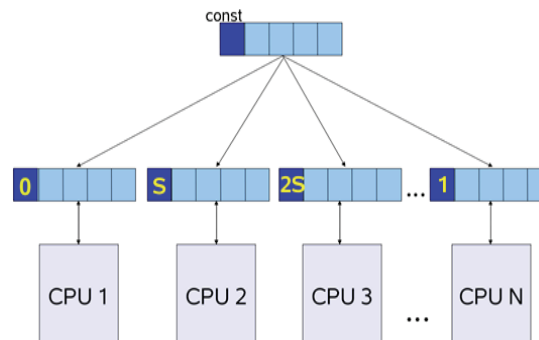


Рис. 1.10. Схема размещения весовых коэффициентов по вычислительным узлам

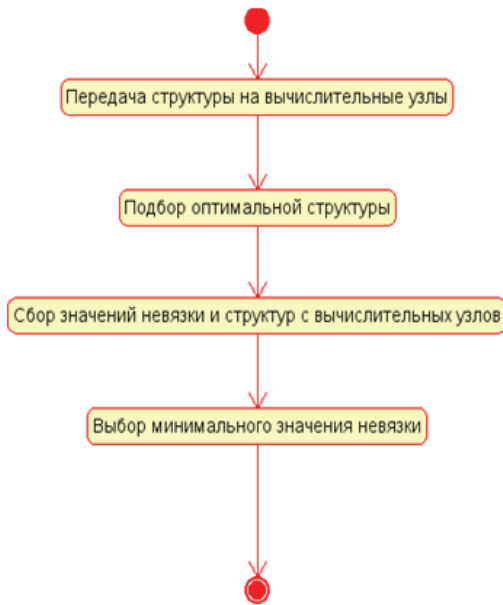


Рис. 1.11. Диаграмма действий подбора весовых коэффициентов методом полного сканирования

шее, соответствующее оптимальной структуре. Диаграмма действий приведена на рис. 1.11.

Распараллеливание на уровне изменения весовых коэффициентов при использовании алгоритма Монте-Карло. Алгоритм подбора весовых коэффициентов Монте-Карло так же, как и другие методы Монте-Карло основан на генерации случайных чисел. В начале значения весовых коэффициентов инициализируются случайным образом, а затем происходит последовательное их уточнение. Происходит это следующим

образом: генерируется случайное число и вычисляется новое значение весового коэффициента:

$$\Delta w_i = s * r;$$

$$w_i = w_i + \Delta w_i,$$

(1.12)

где r – сгенерированное случайным образом число.

После этого вычисляется значение погрешности при новом значении веса, и если разница между текущей и вычисленной погрешностью не отрицательна, то весовой коэффициент не меняется; в противном случае, он возвращается к предыдущему состоянию:

$$w_i = w_i - \Delta w_i.$$

(1.13)

Таким образом, на каждой итерации происходит уточнение всех весовых коэффициентов. Подбор завершается, когда будет выполнено условие остановки (достигнута заданная погрешность, пройдено определенное количество итераций и т. д.).

Модификация метода заключается в том, что для каждого веса генерируется одно, а n случайных чисел $r = (r_0, r_1, \dots, r_{n-1})$, вычисляются

$$w_i = (w_{i,0}, w_{i,1}, \dots, w_{i,n-1}), w_{i,j} = w_i + s * r_j, j = 0, \dots, n -$$

-1 , а затем выбирается минимальный $\varepsilon(w_i)$, и если это значение меньше предыдущего значения целевой функции, то соответствующий ему w_i становится новым значением весового коэффициента.

Каждый процессор независимо генерирует r_j , вычисляет $w_{i,j}$ и $\varepsilon_{i,j}$ и посылает их на ведущий. Ведущий процессор сравнивает $\varepsilon_{i,j}$ и, в зависимости от результата, устанавливает новое значение w_i и посылает его на все вычислительные узлы (рис. 1.12).

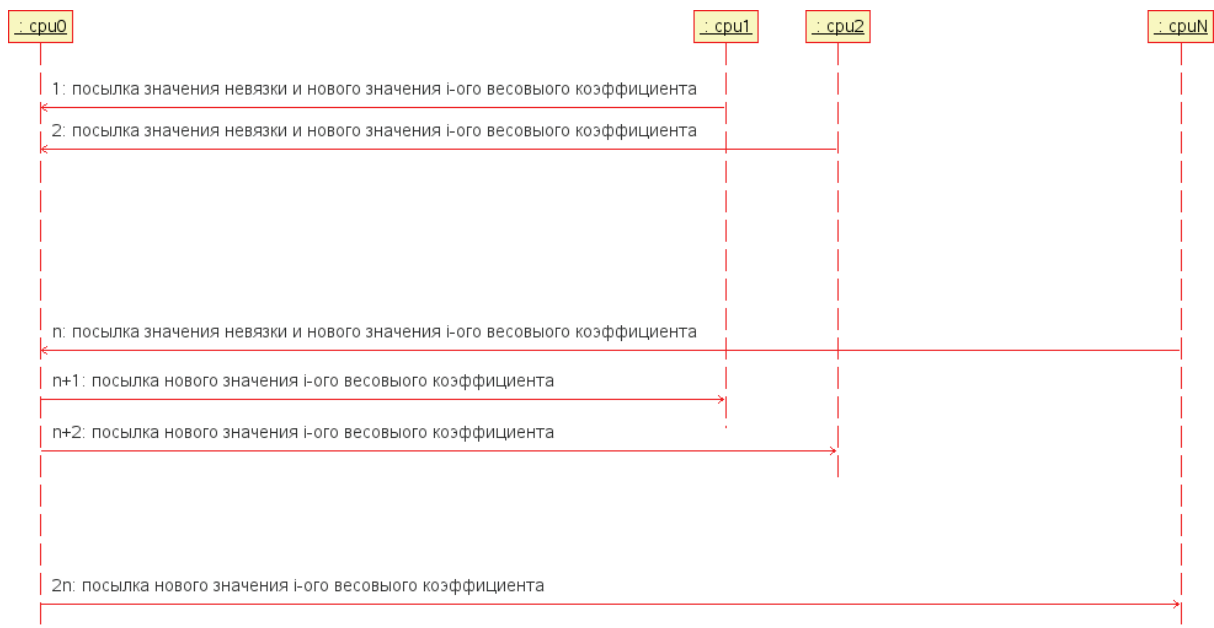


Рис. 1.12. Диаграмма последовательности подбора i -го весового коэффициента на одной итерации

Распараллеливание на уровне изменения весовых коэффициентов при использовании градиентного алгоритма. Градиентный метод подбора весовых коэффициентов лежит в основе многих других очень распространенных методов. Суть его заключается в вычислении вектора градиента и изменения значений весовых коэффициентов в направлении антиградиента. Различные варианты градиентного метода по-разному вычисляют направление антиградиента, например, метод наискорейшего спуска использует формулу (1.14), а QuickProp – (1.15):

$$w_i = w_i + s \frac{\partial \varepsilon(w)}{\partial w}; \quad (1.14)$$

$$w_i = w_i + \begin{cases} \min\left(\frac{g(k)_i}{g(k-1)_i - g(k)_i}, q_{MAX}\right) \Delta w_i, & \Delta w_i \neq 0, \\ sg(k)_i & \Delta w_i = 0. \end{cases} \quad (1.15)$$

Здесь $g(k)_i = \frac{\partial \varepsilon(w)}{\partial w_i}$ – элемента градиента на k -й итерации; q_{MAX} – максимальное значение коэффициента момента.

Как можно заметить, одно из главных отличий этого метода от метода Монте-Карло заключается в том, что для вычисления нового значения одного из весовых коэффициентов используются старые значения остальных весовых коэффициентов, т. е. те, которые были на предыдущей итерации. Исходя из этого, можно сделать вывод, что элементы вектора градиентов могут быть вычислены одновременно, следовательно, этот вектор может быть разделен на n частей, каждая из которых вычисляется на отдельном процессоре (для этого процессору необходимо лишь передать текущие значения весовых коэффициентов). После окончания вычисления процессоры не возвращают полученные

результаты на ведущий, а изменяют значения приписанных к ним весовых коэффициентов, а уже после этого возвращают результат (новые весовые коэффициенты) [34].

Тестирование разработанных механизмов параллельного обучения интеллектуального ядра экспертной системы. Для тестирования разработанных программных механизмов был проведен ряд вычислительных экспериментов, одним из которых являлось «Прогнозирование температуры воздуха в городе Тамбове». Исходные данные для проведения вычислительного эксперимента были предоставлены ГУ ТАМБОВСКИЙ ЦГМС (Государственное учреждение «Тамбовский областной центр по гидрометеорологии и мониторингу окружающей среды»). Они представляли собой файлы, в которых фиксировались дата, время и значения температур воздуха в четырех различных точках г. Тамбова (посты № 1–4). Периодичность замеров составляла 6 часов. Данные покрывали период с января 2000 г. по август 2005 г. На основе имеющихся данных

были сформированы 4 временных ряда – $\vec{T}_1, \vec{T}_2, \vec{T}_3, \vec{T}_4$ (по одному для каждого из постов), длина каждого из них составляет 20454. Для проведения эксперимента был сформирован ряд $\vec{T}$, представляющий собой результат усреднения исходных временных рядов (т. е. каждый i -й элемент сформирован по формуле (1.16)).

$$T_i = \frac{\hat{T}_{1i} + \hat{T}_{2i} + \hat{T}_{3i} + \hat{T}_{4i}}{4}. \quad (1.16)$$

Для нахождения оптимальной архитектуры ИНС в ходе вычислительного эксперимента применялся конструктивный подход, т. е. за основу взята сеть минимального размера, которая постепенно увеличивалась до достижения требуемой точности, при этом на каж-

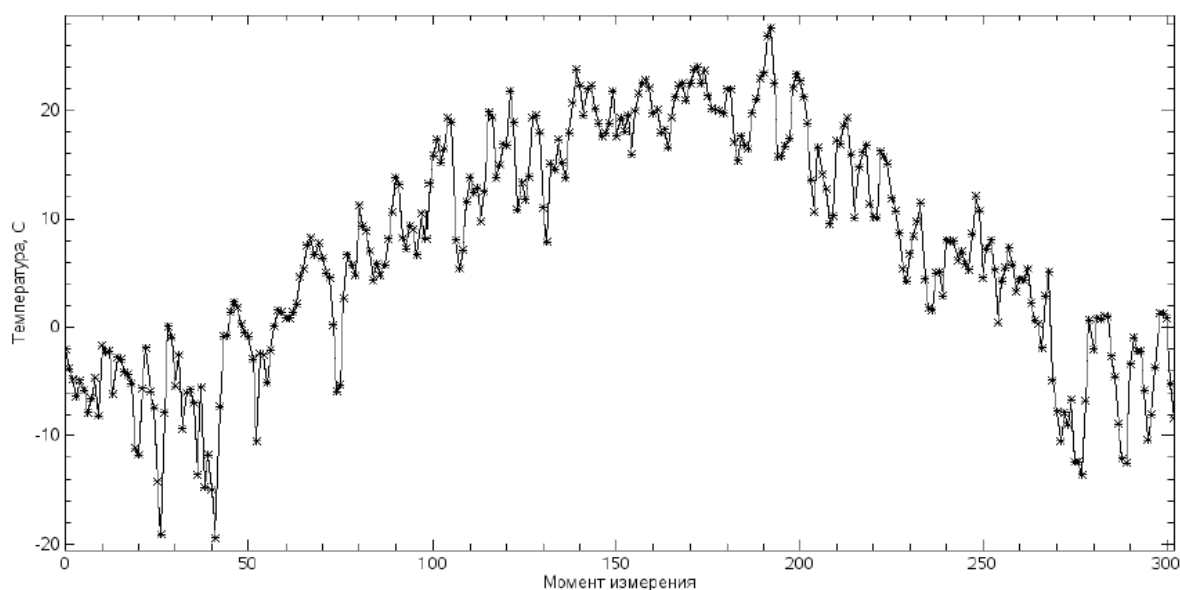


Рис. 1.13. Температура воздуха в городе Тамбове в 2004 году

дом шаге производился подбор активационных функций и весовых коэффициентов.

Начальная конфигурация ИНС изображена на рис. 1.14. Как можно видеть, сеть содержит 1 скрытый слой с 1 нейроном. Все нейроны имеют сигмоидальную активационную функцию. Обучающая выборка была задана следующим образом: количество входов L было взято равным 5, а значение коэффициента смещения $c = 1$. После этого была взята структура, соответствующая наименьшему значению невязки, т. е. структура, имеющая 13 нейронов на скрытом слое, и добавлен еще один слой, содержащий изначально один нейрон (рис. 1.15). Далее последовательно добавлялись новые нейроны, и после каждого изменения структуры производился подбор весовых коэффициентов.

Обучение ИНС производилось с использованием персонального компьютера и кластерной системы. Полученные значения коэффициента эффективности приведены в табл. 1.1.

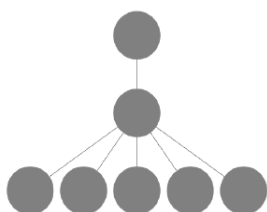


Рис. 1.14. Начальная конфигурация ИНС

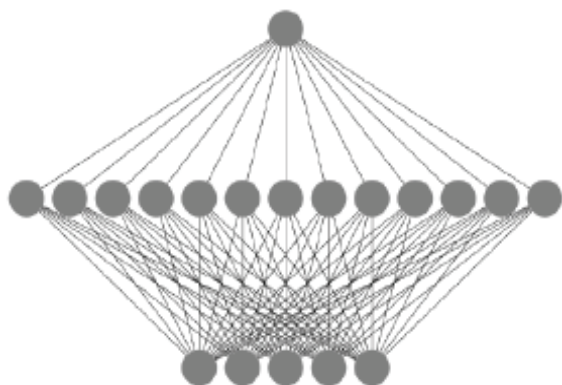


Рис. 1.15. Структура ИНС, соответствующая наименьшему значению невязки

Таблица 1.1

Значения коэффициента эффективности для задачи прогнозирования температуры воздуха

	Кластер ТГУ	Кластер ТГТУ	Кластер МВЦ
α_{wG} (4 процессора)	0.9235	0.9199	0.9297
α_{wG} (6 процессоров)	0.9221	0.9157	0.9291
α_{wG} (8 процессоров)	0.9204	0.9105	0.9283
α_{ε} (4 процессора)	0.8921	0.8822	0.9091
α_{ε} (6 процессоров)	0.8906	0.8793	0.9079
α_{ε} (8 процессоров)	0.8891	0.8769	0.9067

Проведенные вычислительные эксперименты позволяют сделать вывод об успешной реализации программных механизмов параллельного обучения интеллектуального ядра экспертной системы.

1.4. Программная реализация комплекса

При выполнении НИОКР использованы компоненты авторов, защищенные свидетельствами о регистрации программ для ЭВМ №: 2010610579, 2008614091, 2008610860, 2007610622, 2009614144, а в процессе его реализации получены новые объекты интеллектуальной собственности, оформленные на ООО «НПК ЭКСПЕРТНЫЕ СИСТЕМЫ» свидетельства о регистрации программ для ЭВМ №: 2012612512, 2012612513.

Необходимо отметить также, что авторы проекта А.А. Арзамасцев, Н.А. Зенкова, О.В. Крючин перед началом его выполнения прошли стажировку по соответствующей тематике в ведущих Европейских университетах Кобленц (Германия) и Тренто (Италия), демонстрировали свои достижения на Европейской выставке-конференции по современным информационным технологиям ICT-2010 в Брюсселе (Бельгия), имеют публикации по данной тематике в российских и зарубежных изданиях [4–6, 61–69]. По данной тематике защищены 2 кандидатские диссертации (Н.А. Зенкова, Недуахин А.В.) и подготовлена 1 кандидатская диссертация (О.В. Крючин). Эти обстоятельства указывают на соответствие разработок НИОКР мировому уровню.

Для написания комплекса был использован язык C++ и библиотека boost. Разработка проводилась на платформе GNU/Linux (openSUSE и Ubuntu), но компоненты комплекса могут быть скомпилированы также для платформ MacOS X и MS Windows (за исключением параллельного нейросетевого симулятора UNS). Для разработки компонента взаимодействия с пользователем UserInteractor использован также язык PHP и инструмент шаблонизации Smarty. Для разработки компонента хранения информации DBInteractor использована СУБД PostgreSQL и библиотека libPQ. Для разработки нейросетевого симулятора использована библиотека MPI, обеспечивающая межпроцессорную передачу данных. Таким образом, при разработке комплекса было использовано исключительно свободно распространяемое ПО (под лицензиями GPL, Apache и т. д.).

В ходе реализации НИОКР были разработаны новые компоненты комплекса.

UserInteractor. Компонент UserInteractor, предназначенный для взаимодействия с пользователем, состоит из двух частей, первая из которых представляет собой набор PHP скриптов. Именно с этим сервером, реализованным на базе Apache, взаимодействует пользователь. Логическая часть реализована на языке PHP, для реализации экранов использована технология шаблонизации Smarty.

Основные пользовательские экраны:

- index.php – форма авторизации, ввод логина и пароля;
- users.php – список зарегистрированных пользователей, доступен для администратора системы. Пользователи отображаются в виде ссылок на страницы «user.php» с параметрами «userId»;
- user.php – отображение информации о пользователе. Возможность правки при передаче дополнительного параметра «edit»;

- objects.php – Список существующих объектов. Объекты отображаются в виде ссылок на страницы «object.php» с параметрами «objectId»;
- object.php – отображение информации об объекте. Возможность правки при передаче дополнительного параметра «edit»;
- createModel.php – создание модели на основе существующего объекта. Выбор параметров модели;
- structure.php – визуальное редактирование существующей структуры модели.

Вторая часть компонента UserInteractor предназначена для взаимодействия с компонентом Manager и реализована на языке C++ с использованием библиотеки boost.

DBInteractor. Компонент DBInteractor, предназначенный для хранения данных, представляет собой высокоуровневую настройку над СУБД PostgreSQL. Он реализован на языке C++ с использованием библиотек boost и lpq (для запросов к СУБД).

База данных состоит из 4 основных таблиц:

- 1) *пользователи (users)*, содержит поля:
 - уникальный идентификатор (id);
 - логин (login);
 - имя (first_name);
 - отчество (middle_name);
 - фамилия (last_name);
 - права (role);
 - электронный адрес (e_mail);
 - город (city);
- 2) *объекты моделирования (simulation_objects)*, содержит поля:
 - уникальный идентификатор (id);
 - название (name);
 - описание (description);
 - число строк выборки, достаточной для начала обучения (data_number);
 - тип (type);
- 3) *ИНС-модели (ann_models)*, содержит поля:
 - уникальный идентификатор (id);
 - название (name);
 - описание (description);
 - конфигурация (value);
 - индекс модели (index);
 - индекс состояния модели (condition_index);
- 4) *обучающая выборка (pattern)*, содержит поля:
 - уникальный идентификатор (id);
 - значение (value);
 - тип (type);
 - индекс строки (row_index);
 - индекс столбца (col_index).

ANNBuilder. Компонент ANNBuilder, предназначенный для построения моделей ИНС, реализован на языке C++ с использованием библиотеки boost.

UNS. Нейросетевой симулятор UNS реализован на языке C++ с использованием библиотеки SCRVL (включающей сокет и потоки Linux, а также инструменты межпроцессорной передачи данных CrVMPI, базирующейся на MPI). Это единственный компонент комплекса, который может работать только под управлением ОС GNU/Linux.

Manager. Компонент Manager представляет собой центральный компонент комплекса. Он реализован на языке C++ с использованием библиотеки boost.

Разработка проводилась в ОС GNU/Linux open-SUSE и Ubuntu в среде KDevelop. Для компиляции кода использован GCC.

2. ДОРАБОТКА И АПРОБАЦИЯ ТЕХНОЛОГИИ И ПРОГРАММНО-ТЕХНОЛОГИЧЕСКОГО КОМПЛЕКСА

Целью данного раздела является доработка программных механизмов программно-технологического комплекса и апробация предлагаемой технологии проектирования экспертных систем на практических задачах, представляющих собой: а) проектирование экспертной системы для промышленного объекта (технологический процесс выработки биомассы на основе отходов производства этанола); б) проектирование экспертной системы для медицинского объекта (прогнозирование результатов хирургического лечения осложненного колоректального рака).

2.1. Отладка программных механизмов и доработка комплекса

Интерфейс пользователя. При тестировании программных механизмов интерфейса пользователя были выявлены следующие ошибки:

- невозможность переподключения пользователя. Дальнейшее тестирование выявило невозможность для клиента постоянного подключения. С целью устранения выявленных ошибок была использована *технология cookies* для хранения идентификатора пользователя в качестве маркера подключения;
- невозможность сохранить идентификатор пользователя на клиентской машине. При дальнейшем тестировании выяснилось, что серверная часть передает идентификатор под другим именем. С целью устранения ошибки были проведены: модификация клиентской части для правильной обработки сообщений; унификация и стандартизация передаваемых сообщений;
- неправильное отображение информационных полей. Для устранения данной ошибки была проведена модификация интерфейса для правильного отображения.

Сервер. Сервер переставал работать при отправке сообщения сразу после прохождения авторизации пользователя. Для исправления этой ошибки был создан метод serverLog(String). Был создан «мутекс» (одноместный семафор, служащий в программировании для синхронизации одновременно выполняющихся потоков) (рис. 2.1), отвечающий за то, чтобы логи выводились в правильной последовательности в консоль. Затем, чтобы увидеть последовательность операций, после каждой из них была вызвана эта функция и просмотрена вся последовательность действий сервера при авторизации. Был создан простой клиент. Этот клиент производил отдельно авторизацию. И после вызова метода авторизации вызывался метод, отправляющий управляющее сообщение для вывода всех пользователей. При тестировании сервер продолжал работу. После чего снова запускался тест на реальных клиентах. На реальных клиентах сервер не работал.

После чего был заново переписан метод авторизации и приема сообщений от пользователей – это не привело к устранению ошибки, в некоторых случаях

```

* Деструктор
*/
UserMediatorServer::~UserMediatorServer()
{
    if (!m_errorCreateConnectionsNest)
    {
        boost::mutex::scoped_lock scoped_lock(mtx_m_connections);
        //освобождение ресурсов занимаемых подключениями и удаление подключений
        std::map<int, UserMediatorConnection*>::iterator map_it = m_connections.begin();
        int buf = 0;
        UserMediatorConnection *bufConn;
        while (map_it != m_connections.end())
        {
            buf = map_it->first;
            bufConn = map_it->second;
            delete bufConn;
            m_connections.erase(buf);
            ++map_it;
        }
        m_connectionsNest->~ConnectionsNest();
        //освобождение памяти занимаемой объектом подключения комплекса
        if (m_complexConnectionFlag)
        {
            delete m_complexConnection;
        }
    }
}

```

Рис. 2.1. Скриншот программного кода при добавлении «мутекса»

```

Exsumserver.cpp строка 380:
serverLog(String("info 385: authed user"));
//lock m_connections
boost::mutex::scoped_lock scoped_lock(mtx_m_connections);
m_connections[connect->idntificator()] = new UserMediatorConnection(connect, this,
    String("user"), users_id.toInt()); //создаем новое подключение
m_authorizations[users_id.toInt()] = connect->idntificator();
connect->sendMessage(message);
}
Exsumserver.cpp строка 506:
//debug
boost::mutex::scoped_lock scoped_lock(mtx_m_connections);
serverLog(String("info: m_connections.size()=")+String::make(m_connections.size())
    +String(" (exsumserver.cpp 510)"));
}
Exsumserver.cpp строка 511:
//mutex mtx_m_connections
boost::mutex::scoped_lock scoped_lock(mtx_m_connections);
= m_authorizations.count(id);
}
Exsumserver.cpp строка 516:
//lock m_connections
boost::mutex::scoped_lock scoped_lock(mtx_m_connections);
m_connections[m_authorizations[id]]->sendMessage(message);
//отправляем сообщение клиенту под номером id
}
Exsumserver.cpp 599:
{
    boost::mutex::scoped_lock scoped_lock(mtx_m_connections);
    serverLog(String("UserMediatorServer::messageFromUser():
        m_connections[id]->userDisconnect();"));
    m_connections[m_authorizations[id]]->userDisconnect();
    serverLog(String("UserMediatorServer::messageFromUser():
        delete m_connections[id];"));
    delete m_connections[m_authorizations[id]];
    serverLog(String("UserMediatorServer::messageFromUser m_connections.erase(id);"));
    m_connections.erase(m_authorizations[id]);
    //m_authorizations.erase(id);
    serverLog(String("UserMediatorServer::messageFromUser(): end delete user
        connection"));
}

```

Рис. 2.2. Скриншот программного кода при добавлении «мутекса» в другие части программного кода

сервер продолжал терять работоспособность. Дальше были просмотрены все обращения к карте авторизаций, ошибок в реализации обнаружено не было. После был поставлен «мутекс» на все строчки, в которых использовалась карта подключений. После этого сервер стал работать стабильно. Обнаружена причина ошибки – одновременное обращение к массиву подключения удаляющего и извлекающего методов. В некоторых случаях сервер не принимал сообщения от пользователей. Тестирование проводилось на нескольких пользо-

вателях, при одновременном подключении. Были выведены все сообщения, приходящие на сервер, параметры всех строк кода, которые выполнялись при данных подключениях и отправке сообщений. Далее было выведено количество элементов карты подключений при создании подключения и при запуске потока этого подключения, принимающего сообщения от пользователей. В результате были получены результаты, показывающие, что в карте подключения элемент создается позже начала приема сообщений от пользователя. Для исправления этой ошибки был создан «мутекс» (рис. 2.2), который блокирует запуск потока приема сообщений от пользователя, пока не создастся элемент подключения в карте подключений, вследствие чего проблема была устранена.

Работа с базой данных. При тестировании компонента, отвечающего за взаимодействие с базой данных, были отлажены следующие механизмы.

1. Возвращение правильных кодов ошибок.

Для исправления в классе «DBMediator» была создана карта, представленная на рис. 2.3.

Эта карта инициализируется в конструкторе класса «DBMediator» (рис. 2.4).

Жирным шрифтом на рис. 2.4 обозначена инициализация карты ошибок.

Во все методы класса «DBMediator» были добавлены возвращаемые значения в соответствии с этими кодами ошибок, рассмотрим возвращение ошибок на примере следующего метода (рис. 2.5).

```

/**
 * Массив ошибок
 */
mutable std::map<int, int> m_errorsArr;

```

Рис. 2.3. Скриншот программного кода «карты», созданной в классе «DBMediator»

```

/**
 * Конструктор. Попытка подключения к базе данных.
 * @param configWord - параметры подключения к базе данных
 * Содержит параметры "login", "password", "dbname", "host", "port"
 */
DBMediator::DBMediator(const ConfigWords configWord)
{
    m_dbname = configWord["dbname"];

    m_dbConnection
    = new PostgresConnection
    (
        configWord["host"],
        configWord["port"].toInt(),
        configWord["login"],
        configWord["password"],
        configWord["dbname"]
    );

    m_errorsArr[PGRES_COMMAND_OK] = 0; //запрос прошел успешно
    m_errorsArr[PGRES_TUPLES_OK] = 0; //запрос прошел успешно
    m_errorsArr[PGRES_EMPTY_QUERY] = PGRES_EMPTY_QUERY; //postgres получил
    пустой запрос проверьте соединение
    m_errorsArr[PGRES_COPY_OUT] = 0;
    m_errorsArr[PGRES_COPY_IN] = 0;
    m_errorsArr[PGRES_BAD_RESPONSE] = PGRES_BAD_RESPONSE;
    m_errorsArr[PGRES_NONFATAL_ERROR] = PGRES_NONFATAL_ERROR;
    m_errorsArr[PGRES_FATAL_ERROR] = PGRES_FATAL_ERROR;
    m_errorsArr[8] = 8; //error remover user
    m_errorsArr[9] = 9; //permission denied
}

```

Рис. 2.4. Скриншот программного кода конструктора класса «DBMediator»

```

* Удаление пользователя
* @param user - идентификатор пользователя
* @return DBResultCore - результат удаления
DBResultCore DBMediator :: removeUser(const int user)
{
    DBResultCore result;
    ErrorCode error = 0;
    if (user == 1) // attempt to remove administrator
    {
        error = 9;
        result.setError
        (
            m_errors.Arr[error],
            String("невозможно удалить аккаунт администратора/impossible to remove the administrator account")
        );
        return result;
    }
    std::string query = "";
    query = "DELETE FROM t_users WHERE id=";
    query += pr_int2stdString(user);
    query += ";";
    if (pr_idInTable(user, "t_users"))
    {
        error = m_dbConnection->strExec(query);
        String errorText = m_dbConnection->lastErrorText();
        if (pr_isQueryError(error))
        {
            m_dbConnection->clear();
            result.setError
            (
                m_errors.Arr[error],
                String("(")+errorText+String(") ошибка при выполнении\n или запроса, проверьте соединение/error when executing \n a query, check connection")
            );
            return result;
        }
        result.setError
        (
            m_errors.Arr[error],
            String("")
        );
        debug(m_dbConnection->lastErrorText());
        m_dbConnection->clear();
        return result;
    }
    error = 8;
    result.setError
    (
        m_errors.Arr[error],
        String("пользователь не существует в базе данных/user does not exist in database")
    );
    return result;
}

```

Рис. 2.5. Скриншот программного кода одного из метода, иллюстрирующего работу механизма возвращения ошибок

Жирным шрифтом на рис. 2.5 обозначено возвращение ошибки с использованием карты `m_errors.Arr[error]`.

Экранирование строк в запросах. Экранирование строк выполняет функция, программный код которой представлен на рис. 2.6.

Экранирование передаваемых данных в запрос добавлено для всех методов класса `DBMediator`, рассмотрим на примере метода `addUser` (рис. 2.7), в коде жирным шрифтом выделено экранирование.

```

* Экранирование строки
* @param str - строка для экранирования
* @return - экранированная строка
*/
std::string DBMediator :: pr_escapeStdString(std::string str)
{
    char escStr[255];
    int cnt = PQescapeString(escStr, str.c_str(), str.size());
    std::string bufStr = escStr;
    return bufStr;
}

```

Рис. 2.6. Скриншот программного кода функции, отвечающей за экранирование строк

```

DBResult<int> DBMediator :: addUser(const User user)
{
    ErrorCode error = PGRES_COMMAND_OK;
    user.identificator();
    if (user.identificator() > 0)
    {
        error = m_dbConnection->strExec
        (
            "SELECT * FROM t_users WHERE id=" +
            pr_int2stdString(user.identificator())
        );
        if (pr_isQueryError(error))
        {
            return pr_returnErrorQuery(error, user.identificator());
        }
        //если пользователь с выбранным id существует
        if (m_dbConnection->queryResult().size() > 0)
        {
            m_dbConnection->clear();
            std::string update_query = "";
            update_query = "UPDATE t_users SET ";
            update_query += "first_name=";
            update_query += pr_escapeStdString
            (
                pr_wstring2string(user.firstName())
            );
            update_query += ", ";
            update_query += "middle_name=";
            update_query += pr_escapeStdString
            (
                pr_wstring2string(user.middleName())
            );
            update_query += ", ";
            update_query += "last_name=";
            update_query += pr_escapeStdString
            (
                pr_wstring2string(user.lastName())
            );
        }
    }
}

```

Рис. 2.7. Скриншот части программного кода метода `addUser`

Таким образом, были протестированы и отлажены программные механизмы пользовательского интерфейса, работы сервера, взаимодействия с базой данных.

2.2. Проектирование экспертной системы для промышленного объекта

Объект представляет собой биотехнологический процесс, в котором осуществляется утилизация отходов производства этанола на мелассной сырьевой основе с выработкой бактериальной биомассы. Детальное описание данного процесса с описанием основных технологических параметров, а также как объекта управления дано в наших работах [70–72].

На рис. 2.8 показана схема данного биотехнологического процесса. Данная технология была реализована на ОАО «Биохим», г. Рассказово, Тамбовская область. Выбор данного объекта для апробации экспертной системы связан с тем, что в нашем распоряжении имеется значительный объем производственных данных по этому процессу за предыдущие годы.

На рис. 2.9 показана упрощенная схема участка производства бактериальной биомассы, являющегося объектом рассмотрения. Основными технологическими единицами являются: биореактор, в котором осуществляется выработка биомассы; термофлотатор, служащий для ее концентрирования, и сушилка-гранулятор, в которой биомасса высушивается и приобретает товарный вид.

Основной технологической единицей, влияющей на утилизацию отходов и выработку биомассы, является четырехсекционный биореактор-ферментер, детальное описание которого также приведено в наших работах [70–72].

Мелассная послеспиртовая барда

Аэрация

Биореактор

Смеситель

Термофлотатор

Отстойник

Окончательная биологическая очистка

Сушилка-гранулятор

Концентрированная биомасса

Продукт

Горячий воздух

Твердый материал

Очищенная жидкость

Поток жидкости

Поток газа

CO₂

CO₂

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

зультатов определения технологических параметров, полученных с использованием экспертной системы и в реальных условиях. Необходимо отметить, что описание технологии удаленного доступа было приведено в предыдущих этапах и потому здесь не приводится.

F	s0	T1	T2	T3	T4	pH1	pH2	pH3	pH4	x1	x2	x3	x4	s1	s2	s3	s4
9,54	29,75	20,63	25,11	27,06	30,40	7,90	7,75	8,70	8,49	10,18	13,78	15,53	15,89	14,47	5,46	1,08	0,20
9,45	29,35	20,14	24,22	28,36	30,56	7,62	7,66	8,39	9,12	8,86	12,50	15,13	15,73	17,50	8,41	1,82	0,33
9,77	29,20	19,73	24,78	28,30	29,61	7,08	8,09	8,95	9,03	6,85	12,02	14,92	15,55	22,10	9,19	1,93	0,36
10,38	30,83	19,38	24,87	27,37	30,33	7,95	8,31	8,69	9,22	8,60	13,52	15,52	15,95	18,91	6,35	1,35	0,26
10,05	29,68	20,02	24,71	28,02	29,71	7,54	7,59	8,30	8,62	7,93	11,48	14,68	15,55	19,58	10,73	2,71	0,54
9,11	30,54	20,87	24,37	28,96	30,30	7,23	8,44	8,85	9,04	8,29	13,96	15,98	16,33	20,30	6,11	1,08	0,18
9,96	31,22	19,42	24,30	28,09	30,23	7,23	8,48	8,14	9,05	7,05	13,10	15,58	16,22	23,29	8,17	1,97	0,36
9,08	28,96	20,39	25,85	27,71	30,00	7,31	8,15	8,43	8,85	8,30	13,61	15,42	15,80	18,94	5,67	1,13	0,19
9,27	30,26	19,53	25,80	27,76	30,58	7,14	7,64	8,26	8,89	7,94	11,87	15,21	16,06	22,26	10,94	2,59	0,46
10,62	29,79	19,25	25,90	27,91	30,89	7,87	7,65	8,49	9,00	7,90	11,72	14,67	15,43	19,30	9,74	2,39	0,48
10,98	28,70	19,08	24,02	27,01	30,44	7,85	7,73	8,66	9,21	7,38	10,67	13,92	14,86	19,36	11,15	3,01	0,67
10,71	29,96	19,58	24,63	27,09	29,25	7,84	8,43	8,33	8,99	7,93	12,79	14,78	15,31	18,78	6,64	1,66	0,34
9,41	29,46	20,87	24,32	28,65	30,40	7,91	7,64	8,89	8,42	10,57	13,69	15,50	15,84	13,36	5,56	1,03	0,19
10,37	30,66	19,59	25,98	27,09	30,20	7,57	7,99	8,74	9,03	7,55	12,63	15,24	15,86	21,14	8,43	1,89	0,36
10,05	30,96	19,94	25,08	28,09	30,60	7,12	8,13	8,65	9,11	6,83	12,39	15,39	16,05	23,44	9,55	2,05	0,39
10,02	31,41	20,62	24,96	27,44	29,56	7,81	7,57	8,44	8,88	9,48	13,03	15,63	16,24	17,25	8,37	1,87	0,34

Рис. 2.10. Фрагмент файла данных наблюдений за биотехнологическим процессом

На рис. 2.10 показан фрагмент файла данных, полученных с технологического процесса (рис. 2.8–2.9). Здесь приведены численные значения входных параметров: расхода отхода производства этанола – F , м³/ч, его концентрации – S_0 , кг/м³ (по БПК), температур в секциях реактора – T_1 – T_4 , °С, кислотности среды в секциях реактора – pH_1 – pH_4 и выходных параметров: концентраций биомассы в секциях – x_1 – x_4 , кг/м³ – и концентраций субстрата в секциях реактора – s_1 – s_4 , кг/м³ (по БПК).

Таким образом, полный файл данных представляет собой прямоугольную матрицу, содержащую 1000 строк, каждая из которых характеризует состояние биореактора, и 18 столбцов, элементы каждого из которых характеризуют один из параметров состояния данного процесса. Необходимо отметить, что все параметры соответствуют схемам объекта (рис. 2.8–2.9).

Построение экспертной системы для данного объекта заключается в разработке математической модели, позволяющей связать между собой входные (возмущения), управляющие и выходные параметры биореактора, заданные их значениями (рис. 2.10), т. е. получить зависимость вида:

$$Y = F(x, u, P). \quad (2.1)$$

Такая модель представляет собой наиболее важную часть экспертной системы – ее решатель, совмещенный в единое целое с базой знаний. Поскольку в данной работе речь идет о разработке экспертной системы, имеющей в качестве интеллектуального ядра модель на основе искусственных нейронных сетей (ИНС-модель), то операторное уравнение (2.1) можно переписать в следующем виде:

$$Y = F_S(x, u, w), \quad (2.2)$$

где x – вектор входных координат (возмущений) объекта, представляющих собой набор входных переменных, на которые пользователь объекта не может оказывать воздействия или влияния; u – вектор варьируемых параметров (управляющих воздействий), который представляет собой набор входных переменных, с помощью которых пользователь объекта может оказывать влияние на процесс, Y – вектор выходных координат, численно выражающий интересующие свойства объекта (рис. 2.11); w – вектор весовых коэффициентов ИНС-модели, размерность которого определяет также ее число степеней свободы для настройки (обучения) по эмпирическим данным (рис. 2.10); F_S – оператор связи, зависящий от структуры ИНС-модели – S .



Рис. 2.11. Схема представления объекта для разработки ИНС-модели

В нашем случае x представляет собой одномерный вектор с двумя координатами F и S_0 , u – одномерный вектор с восемью координатами T_1 – T_4 и pH_1 – pH_4 , Y – одномерный вектор с восемью координатами x_1 – x_4 и s_1 – s_4 .

Полученная модель должна быть нелинейной, т. к., например, в нее в качестве входных величин входят pH (показатель кислотности) среды, являющийся логарифмической функцией концентрации ионов водорода и температура, зависимость от которой основных показателей биотехнологического процесса выражается экспоненциальным уравнением Аррениуса [70].

По данным технологического процесса (рис. 2.10) разрабатывали экспертную систему с использованием нашего программного обеспечения вначале с использованием одного процессора, а затем с использованием вычислительного кластера. Ниже приведены результаты данной работы.

Так, на рис. 2.12 и 2.13 показан процесс обучения ИНС-модели с определенной конфигурацией. В данном случае речь идет об обучении ИНС с десятью входными нейронами, тремя нейронами в скрытом слое (линейная, квадратичная и кубическая функции) с использованием одного единственного процессора градиентным методом. Из рис. 2.12 и 2.13 видно, что погрешность модели быстро уменьшается, т. е. сеть с такой конфигурацией быстро обучается эмпирическим данным.

Необходимо отметить, что разрабатываемое нами программное обеспечение включает также авторский конструктивный алгоритм выбора структуры сети, описанный нами в [4, 6, 61–62, 73].

Тот факт, что в логарифмической системе координат (рис. 2.13) эмпирические данные по обучению ИНС-модели достаточно хорошо ложатся на прямую линию (с коэффициентом корреляции 0,983), указывает на возможность использования для описания данного процесса зависимостей вида (1.5)–(1.7).

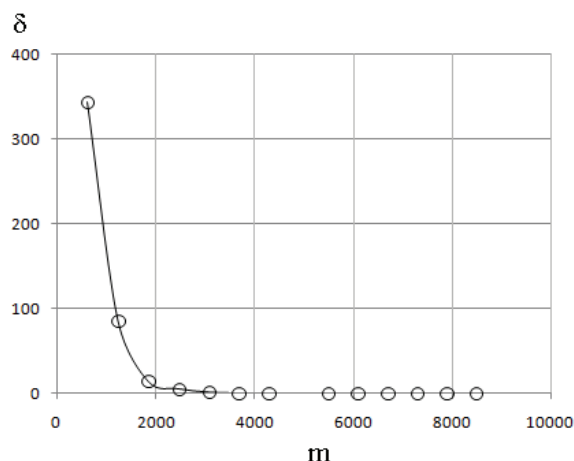


Рис. 2.12. Процесс обучения ИНС-модели (линейная, квадратичная и кубическая функции нейронов) градиентным методом: δ – среднеквадратическая погрешность, m – число тактов обучения. Один процессор

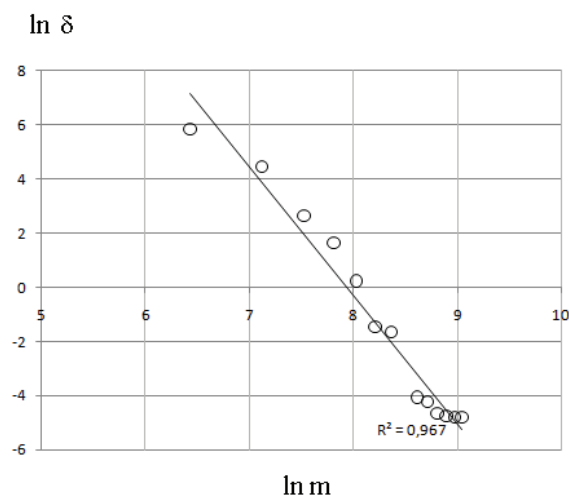


Рис. 2.13. Процесс обучения ИНС-модели (линейная, квадратичная и кубическая функции нейронов) градиентным методом. Один процессор. В логарифмических координатах

Предложенные уравнения на практике позволяют решить следующие задачи: оценить первичный период накопления информации, необходимой для построения ИНС-модели исследуемого объекта; определить начальную структуру ИНС-модели на основе входных характеристик объекта исследования; сделать оценку количества циклов обучения ИНС-модели в присутствии эксперта.

На рис. 2.14–2.17 показаны корреляции эмпирических данных (рис. 2.10) и значений x_1 , рассчитанных по ИНС-модели для различных структур ИНС. Из этих рисунков следует, что наилучшая структура ИНС-модели, соответствующая эмпирическим данным и определенная по среднеквадратической погрешности: десять входных нейронов, три нейрона с линейной, квадратичной и кубической передаточными функциями в скрытом слое и один выходной нейрон.

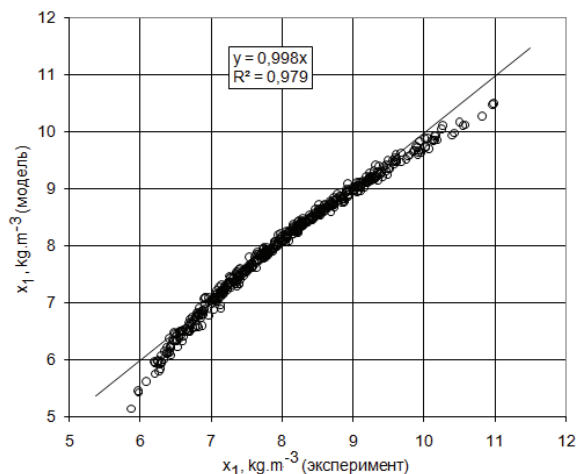


Рис. 2.14. Корреляция реальных данных – x_1 и расчетов x_1 по ИНС-модели для нейронной сети с одним линейным нейроном. Изображена корреляционная зависимость для 500 первых данных

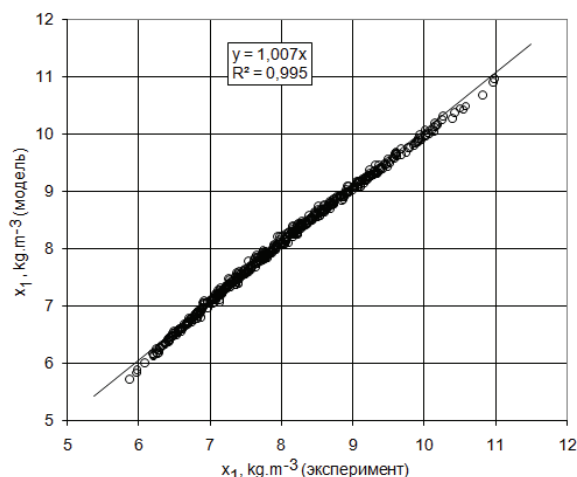


Рис. 2.15. Корреляция реальных данных – x_1 и расчетов x_1 по ИНС-модели для нейронной сети с одним линейным и одним квадратичным нейронами. Изображена корреляционная зависимость для 500 первых данных

В табл. 2.1 приведены основные параметры моделей, корреляции для которых приведены на рис. 2.14–2.17.

На рис. 2.18 приведена также и схема ИНС-модели. Обучение ИНС-моделей, используемых в качестве интеллектуального ядра экспертных систем, проводили и на вычислительных кластерах. При этом проводили расчеты концентраций субстрата в первой и четвертой секциях и концентрации биомассы в четвертой секции реактора: S_1 , S_4 и X_4 .

Был использован вычислительный кластер Тамбовского государственного технического университета, имеющий 4 процессора. Распараллеливание осуществляли на уровне вычисления значения целевой функции. Эффективность составили 90–92 %, время расчетов параллельной версии – около 4 часов с автоматизированным поиском оптимального варианта структуры

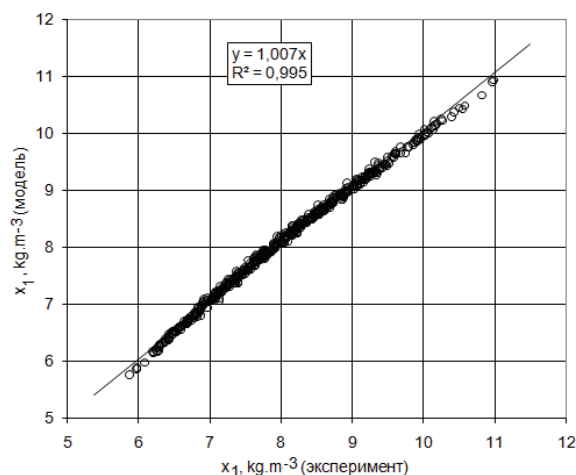


Рис. 2.16. Корреляция реальных данных – x_1 и расчетов x_1 по ИНС-модели для нейронной сети с одним линейным, одним квадратичным и одним кубическим нейронами. Изображена корреляционная зависимость для 500 первых данных

ИНС-модели, показанной на рис. 2.19. В качестве оптимальной структуры ИНС-модели кластерной версией программы автоматизировано выбрана структура с одним кубическим, двумя квадратичными и четырьмя линейными нейронами. При этом относительные погрешности составили: для x_4 – 9 %, для s_1 – 19 %. для s_4 – 2 %. Корреляции эмпирических данных и расчетов

по ИНС-модели для этих случаев показаны на рис. 2.20, 2.21 и 2.22.

Сравнение времени выполнения программы на единичном процессоре и вычислительном кластере показало, что существенная доля временных затрат получения ИНС-модели экспертной системы приходится на поиск структуры сети.

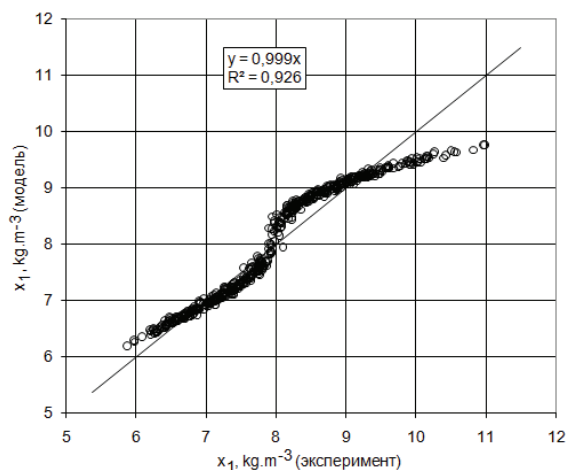


Рис. 2.17. Корреляция реальных данных – x_1 и расчетов x_1 по ИНС-модели для нейронной сети с одним сигмоидальным нейроном. Изображена корреляционная зависимость для 500 первых данных

Таблица 2.1

Основные показатели описания эмпирических данных биотехнологического процесса с использованием различных ИНС-моделей, используемых в качестве интеллектуального ядра экспертных систем

Характеристика скрытого слоя	Коэффициент корреляции расчетных и эмпирических данных	Средняя квадратическая ошибка ИНС-модели	Время обучения ИНС-модели, мин.
Один линейный нейрон	0,989	0,02178	1,5
Линейный и квадратичный нейроны	0,997	0,00837	3
Линейный, квадратичный и кубический нейроны	0,997	0,00826	4,3
Сигмоидальный нейрон	0,962	0,07659	3

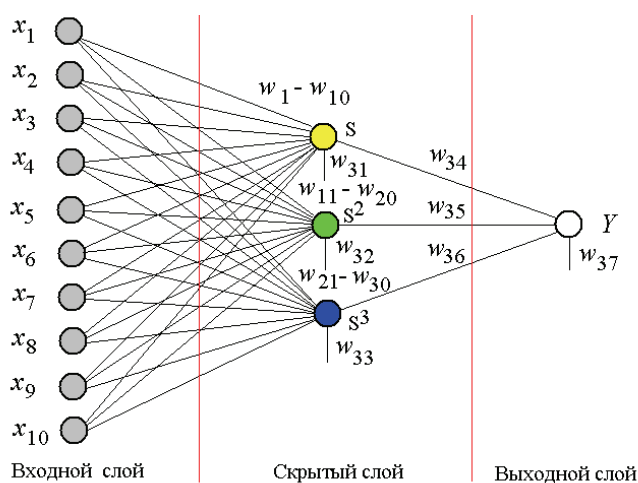


Рис. 2.18. Структура ИНС-модели, обеспечивающей адекватное представление эмпирической информации для производственного процесса, представленного на рис. 2.8 и 2.9

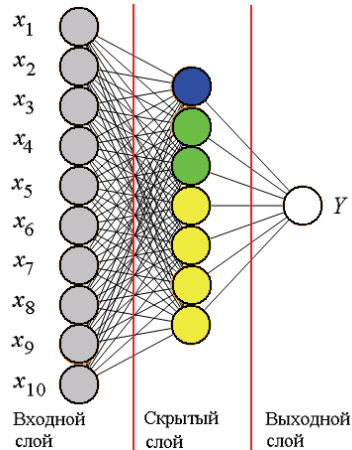


Рис. 2.19. Структура ИНС-модели, полученная с помощью вычислительного кластера и обеспечивающая адекватное представление эмпирической информации для производственного процесса, представленного на рис. 2.8 и 2.9

Весовые коэффициенты ИНС-модели (рис. 2.19) для расчета s_1 приведены ниже:

0,4288162291 0,6911659837 0,1684842408
 0,4616163075 0,7723379135 0,4216362238
 0,9884165525 0,9788873196 0,6507799029
 0,9671633244 0,5300018191 0,01564698666
 0,3375403881 0,2216714323 0,4493710399
 0,7133677006 0,5164060593 0,06919586658
 0,4069945812 0,370903194 0,505251348 0,1388458759
 0,4709280729 0,6744026542 0,6213706732
 0,07822722197 0,8713401556 0,3879356384
 0,5161554813 0,8892344236 0,1537122428
 0,1915409863 0,5621814728 0,5658055544
 0,9058300853 0,5828891993 0,995773077 0,9083153009
 0,8174400926 0,6529616714 0,1239705533
 0,3195488155 0,9493271112 0,4367306828
 0,8182460666 0,6685219407 0,4569704831
 0,3795740008 0,9782751203 0,8832216859
 0,9629584551 0,4627134502 0,2770597935
 0,4101133347 0,1543281227 0,9040151834
 0,7805030942 0,3157556355 0,2891126871
 0,5657803416 0,1679721922 0,4115138948
 0,7431307435 0,7208458185 0,267467916 0,8945841193
 0,5835685134 0,2519495189 0,07455415279
 0,384468019 0,1137303263 0,3678540885 0,9564149976
 0,01722002774 0,8362314701 0,78604424
 -0,2695320249 0,5538398623 0,4061145186
 -0,01418774575 0,673956871 0,5876576304
 0,4519741535 0,9328891635 0,04772821069
 0,8873259425 0,1167072952 0,8477321267
 0,2128166556 0,4433260858 0,3933708966 0,396352917
 0,1390984654 0,3976278305 0,375639379

Весовые коэффициенты ИНС-модели (рис. 2.19) для расчета s_4 :

-0,5704417825 -0,3067550063 -0,8308624625
 0,4633195698 0,7747497559 -0,5769819021
 0,9883269072 -0,02421592176 0,651948154
 -0,03596661985 -0,4777849317 0,03908353299
 -0,6614277363 -0,7468106151 -0,5182107091
 -0,2548091412 -0,479221344 -0,9636934996
 -0,5867102146 -0,6539791226 0,4907354712

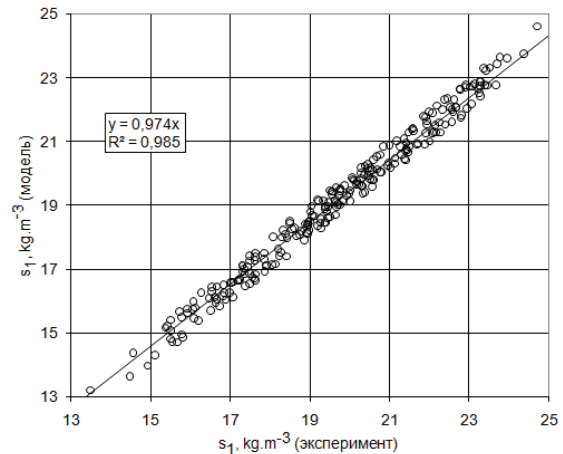


Рис. 2.20. Корреляция реальных данных – s_1 и расчетов s_1 по ИНС-модели для нейронной сети, показанной на рис. 2.19. Изображена корреляционная зависимость для 250 первых данных

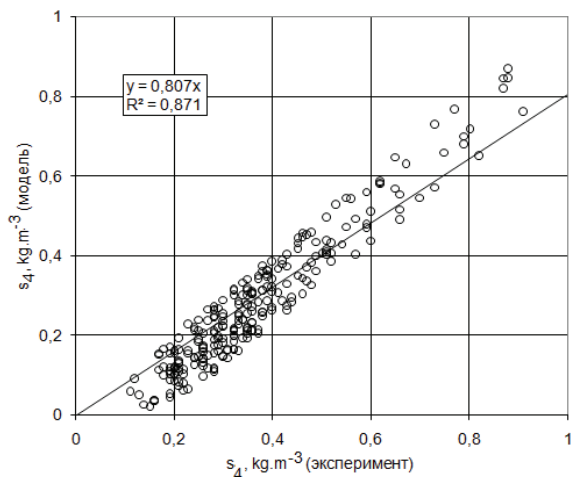


Рис. 2.21. Корреляция реальных данных – s_4 и расчетов s_4 по ИНС-модели для нейронной сети, показанной на рис. 2.19. Изображена корреляционная зависимость для 250 первых данных

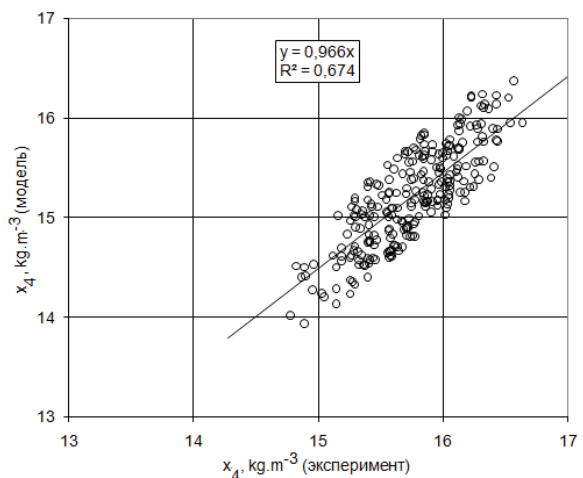


Рис. 2.22. Корреляция реальных данных – x_4 и расчетов x_4 по ИНС-модели для нейронной сети, показанной на рис. 2.19. Изображена корреляционная зависимость для 250 первых данных

–0,842656672 0,4675816298 –0,3012738824 0,647780776
 –0,8964956403 0,8684352636 0,3658132553
 –0,4878232479 –0,1296569407 –0,9198666215
 –0,5133520961 0,5029249191 0,5552321672
 0,1379725337 –0,2230418921 0,8237388134
 –0,05016455799 –0,3082303703 0,6203956008
 0,110185191 0,3382017314 0,9468284845 0,4570706487
 –0,1558273286 0,6927443147 0,4551470578
 0,3587762713 –0,02263088711 0,86644274
 0,0193118006 –0,520377934 0,250112772 –0,0408744812
 0,1773202568 –0,2229760885 0,6667154431
 –0,5945608616 –0,5990222692 0,5554787517
 0,1193605512 –0,3891767859 0,7373982072
 –0,3085044324 0,384406507 0,911829412 –0,4308186471
 0,2650654018 –0,7632086873 0,3955869079
 –0,8295327425

Весовые коэффициенты ИНС-модели (рис. 2.19) для расчета x_4 :

–0,5694060922 –0,307300508 –0,8305162787
 0,462882787 0,7741994858 –0,5763953924 0,9889353514
 –0,02372954786 0,6522523761 –0,03583501279
 –0,381878674 –0,01728148013 –0,636954546
 –0,7451941371 –0,5691607594 –0,2912786007
 –0,4150056839 –0,9485729933 –0,5316421986
 –0,6321180463 0,4832786918 –0,8323836923
 0,4667853117 –0,3034873605 0,6527198553
 –0,8909805417 0,8609946966 0,3653144836
 –0,492610693 –0,1329366267 –0,8492556214
 –0,5841318965 0,500246048 0,5419026613
 0,06970185786 –0,2706416845 0,8380429745
 –0,1023810878 –0,2908897102 0,5919341445
 0,1276708096 0,3295662105 0,9494491816
 0,4543088078 –0,1653430909 0,6837120652
 0,4659579098 0,3602868915 –0,01319523342
 0,8676882386
 –0,03200615942 –0,4333055615 0,1920378208
 –0,07105731964 0,1847007424 –0,138266921
 0,6991356015 –0,6002905369 –0,6089967489
 0,5872494578 0,1409412473 –0,4160847068
 0,7910585999 –0,2642692029 0,3661999106
 0,9121851325 –0,4759331644 0,2709161937
 –0,812210381 0,3853225112 –0,8021785021

Таким образом, проектирование экспертной системы для реального объекта (технологического процесса выработки биомассы), согласно условиям поставленной практической задачи, показало применимость и реализуемость предлагаемых нами подходов.

2.3. Проектирование экспертной системы для медицинского объекта

В качестве второго примера, для которого была реализована экспертная система по технологии, предлагаемой в данном проекте, рассмотрен медицинский объект.

Задача разработки экспертной системы ставилась следующим образом. Необходимо спроектировать экспертную систему для прогнозирования исхода (выздоровление пациента и летальный исход) хирургического лечения осложненного колоректального рака.

Экспертная система должна базироваться на эмпирических данных лечения 159 реальных больных за

период 2004–2012 гг., истории болезни больных и результатах анализов.

Перечень входных параметров экспертной системы и их шифровка приведены ниже.

1. Возраст больного

До 40 лет – 1
 От 41 до 50 лет – 2
 От 51 до 60 лет – 3
 От 61 до 70 лет – 4
 От 71 до 80 лет – 5
 Старше 80 лет – 6

2. Пол больного

Мужской – 1
 Женский – 2

3. Наличие сопутствующей патологии

3.1. Нарушение мозгового кровообращения

1 – Да
 0 – Нет

3.2. Почечная недостаточность

1 – Да
 0 – Нет

3.3. Перенесенный острый инфаркт миокарда

1 – Да
 0 – Нет

3.4. Сердечная недостаточность

1 – Да
 0 – Нет

3.5. Дыхательная недостаточность

1 – Да
 0 – Нет

3.6. Сахарный диабет

1 – Да
 0 – Нет

3.7. Другие

1 – Да
 0 – Нет

4. Пульс

Менее 60 в минуту – 0
 Норма 60–90 в минуту – 1
 От 91 до 100 в минуту – 2
 От 101 до 120 в минуту – 3
 Свыше 120 в минуту – 4

5. Длительность заболевания

От 4 до 6 часов – 1
 От 6 часов до 12 часов – 2
 От 12 часов до суток – 3
 От суток до трех – 4
 От трех до 7 суток – 5
 Более 7 суток – 6

6. Систолическое артериальное давление

Ниже 90 – 0
 От 91 до 110 – 1
 От 110 до 140 – 2
 От 141 до 180 – 3
 Выше 180 – 4

7. Частота дыхательных движений

Менее 13 в минуту – 0
 13–18 в минуту – 1
 От 19 до 25 в минуту – 2
 Выше 25 в минуту – 3

8. Лейкоциты крови

Менее 4 – 0
 От 4 до 9 – 1
 От 9,1 до 12 – 2

Более 12 – 3

9. Сдвиг лейкоцитарной формулы влево

9.1. Ядерный индекс интоксикации

0,05 – 0,08 – 1

От 0,09 до 0,29 – 2

0,3 – 1,0 – 3

Более 1,0 – 4

10. Наличие метастазов

Отсутствуют – 1

Региональные лимфоузлы – 2

Отдаленные лимфоузлы и отдаленные органы – 3

11. Размеры кишки

11.1. Толстая кишка

Не раздута – 1

Раздута умеренно – 2

Раздута значительно – 3

11.2. Тонкая кишка

Не раздута – 1

Раздута умеренно – 2

Раздута значительно – 3

12. Длительность операции

До часа – 1

От 1 часа до 2-х часов – 2

От 2-х до 3-х часов – 3

Свыше 3-х часов – 4

13. Опухоль прорастает

Не прорастает окружающие структуры – 0

Забрюшинную клетчатку – 1

Соседние органы – 2

Крупные сосуды – 3

Мочеточники – 4

14. Наличие осложнений:

Без осложнений – 0

Перитонит – 1

Абсцессы – 2

Некроз кишки – 3

Перфорация – каловый перитонит – 4

15. Рвота

Да – 1

Нет – 0

16. Наличие выпота в брюшной полости

Да – 1

Нет – 0

17. Объем операции:

Радикальная – по типу обструктивной резекции – 1

Паллиативная – наложение колостомы – 2

18. Исход

Выписан, выздоровление – 1

Летальный исход – 2.

Разработка экспертной системы для прогнозирования результатов хирургического лечения осложненного колоректального рака. После преобразования исходных данных, предоставленных заказчиком, получен файл, фрагмент которого представлен на рис. 2.23.

Файл содержит 156 строк данных, каждая из которых соответствует индивидуальному случаю (больному), и 25 столбцов, первые 24 – это входы изучаемого объекта (x_1 – x_{24} , см. файл), и 25-й столбец – выход объекта (Y). Фамилии пациентов из файла исключены по этическим соображениям.

Репрезентативность данных не может быть объективно проверена, т. к. для этого была бы необходима выборка следующего объема (число строк): $2^{24} = 16777216$. Этот расчет сделан в предположении, что каждый вход имеет только два состояния (0 и 1) и для оценки репрезентативности нужна лишь одна запись. Реально это число должно быть увеличено еще в 10–20 раз.

Для предварительной обработки данных использовали симулятор ИНС, описанный в наших публикациях [5, 7, 63–65, 69], и свидетельства о государственной регистрации программ.

Наращивание структуры сети, как правило, производят от простого к сложному до тех пор, пока не будет достигнута удовлетворительная погрешность генерализации эмпирических данных.

По этой причине на первом этапе обучали нейронную сеть, имеющую 24 входа, один скрытый слой, включающий один линейный нейрон и один выходной слой, также содержащий один нейрон. Необходимо отметить, что такая линейная аппроксимация имеет особый смысл, т. к. в результате получается по сути дела линейное приближение ряда Тейлора:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24	Y
1	2	3.1	3.2	3.3	3.4	3.5	3.6	3.7	4	5	6	7	8	9.1	10	11.1	11.2	12	13	14	15	16	17	18
3	1	0	0	0	0	0	0	0	2	3	2	2	1	2	1	2	1	1	0	0	1	0	1	1
4	2	0	0	0	0	1	0	1	1	4	2	1	2	1	1	2	1	4	0	0	0	0	1	1
4	1	0	0	0	0	0	0	1	1	5	2	0	1	4	1	2	1	1	0	0	0	1	0	1
5	2	0	0	0	1	0	0	1	1	3	2	1	2	2	1	2	1	1	0	0	1	1	1	1
6	2	0	0	0	0	1	0	0	2	5	3	1	1	2	1	2	2	2	0	0	0	0	1	1
4	1	0	0	0	0	0	0	1	2	5	2	1	2	2	1	2	2	2	0	0	0	0	1	1
4	2	0	0	0	0	0	1	1	2	5	3	1	1	2	1	2	1	3	0	0	0	0	1	1
5	2	0	0	0	0	0	0	1	1	3	2	1	1	1	1	2	1	2	0	0	0	0	1	1
4	1	0	0	0	0	0	0	1	1	4	2	1	1	3	1	3	2	2	0	0	0	0	1	1
3	1	0	0	0	0	0	0	0	3	2	2	1	3	3	3	2	1	1	0	0	1	0	1	1
3	1	0	0	1	1	0	0	0	1	5	2	1	1	1	1	2	1	2	0	0	0	0	1	1
5	2	0	0	0	1	0	1	0	4	5	3	1	3	3	3	2	1	3	2	1	0	1	2	1
5	1	0	0	1	1	0	0	1	1	4	2	2	0	1	1	2	1	2	0	0	1	0	2	1
5	2	0	0	0	1	0	0	1	3	5	3	2	3	3	1	3	2	2	0	0	0	0	2	2
1	1	0	0	0	0	0	0	0	2	4	2	1	3	1	1	2	2	4	0	0	0	1	1	1
4	2	0	0	0	0	0	0	0	1	5	3	1	1	1	1	2	1	2	0	0	0	0	2	1
4	1	0	0	0	0	0	0	0	1	4	2	1	1	2	3	2	1	1	2	0	0	1	2	1
6	2	0	0	0	1	0	1	1	2	4	3	1	1	1	3	3	1	1	0	0	0	1	2	1
4	2	0	0	0	0	0	0	1	2	4	3	2	1	2	3	2	1	2	2	1	0	1	2	1
5	2	0	0	0	1	0	0	1	1	1	3	1	1	2	1	2	1	2	0	0	0	0	1	1
5	1	0	0	1	1	0	0	1	1	3	2	1	0	2	1	2	1	2	0	0	1	0	2	1
6	2	0	0	0	1	0	0	1	1	5	2	1	1	1	1	2	1	2	0	0	0	0	2	1
6	2	0	0	0	1	0	1	1	3	5	3	2	1	1	3	3	2	2	0	0	0	1	2	2
5	1	0	0	0	0	0	0	1	4	3	0	2	1	4	3	3	2	4	0	4	1	1	2	1
1	1	1	0	0	0	1	0	1	1	5	3	1	2	1	1	2	1	1	0	0	1	0	2	1

Рис. 2.23. Фрагмент файла исходных данных для разработки экспертной системы. Представлены входные параметры x_1 – x_{24} и исход Y

$$\begin{aligned}
f(\mathbf{x}) &= f(a) + \frac{\partial f(\mathbf{x})}{\partial x_i} (x_i - a) + \\
&+ \frac{1}{2!} \frac{\partial^2 f(\mathbf{x})}{\partial x_i^2} (x_i - a)^2 + \dots + \\
&+ \frac{1}{(n-1)!} \frac{\partial^{(n-1)} f(\mathbf{x})}{\partial x_i^{n-1}} (x - a)^{n-1} + O_n(x),
\end{aligned} \quad (2.3)$$

по которому легко оценить чувствительность каналов:

$$f(\mathbf{x}) = f(a) + \frac{\partial f(\mathbf{x})}{\partial x_i} (x - a) + O_2(x), \quad (2.4)$$

где $f(\mathbf{x})$ – функция нескольких переменных; a – рабочая точка; i – порядковый номер рассматриваемой переменной.

При обучении погрешность ИНС-модели быстро снижается за счет изменения весовых коэффициентов. Среднеквадратичная погрешность обучения составила 0,08, количество случаев неверного обучения – 12 из 156, что нельзя считать приемлемым результатом.

На диаграмме (рис. 2.24) показана чувствительность выхода – исход (выписан, выздоровление или летальный исход) – к входным параметрам. Из этой диаграммы можно сделать вывод о том, что исход крайне малочувствителен к возрасту больного, нарушениям мозгового кровообращения, сахарному диабету и другим сопутствующим патологиям. С другой стороны, наиболее значимыми факторами с точки зрения исхода являются: почечная недостаточность, сердечная недостаточность, частота дыхательных движений, наличие метастазов, размеры толстой кишки, наличие осложнений. Влияют на исход такие факторы, как пол больного, перенесенный острый инфаркт миокарда, дыхательная недостаточность, пульс, длительность заболевания, систологическое артериальное давление, лейкоциты крови, ядерный индекс интоксикации, длительность операции, прорастание опухоли, рвота, наличие выпота в брюшной полости и объем операции прослеживается на среднем уровне.

Необходимо отметить, что рис. 2.24 позволяет оценить и направление влияния фактора. Например, чувствительность фактора 2 (пол больного) является отрицательной, что в силу принятого шифрования означает, что женщины чаще выздоравливают, чем мужчины. Вероятность летального исхода возрастает при почеч-

ной и сердечной недостаточности, высокой частоте дыхательных движений, высоком уровне лейкоцитов крови, наличии метастазов, повышенного размера толстой и тонкой кишки, наличии осложнений. В то же время вероятность летального исхода снижается при повышении пульса и ядерного индекса интоксикации, увеличении длительности операции, смещению объема операции в сторону паллиативной.

Следует отметить также, что вход 24 (объем операции) в силу объективных причин является управляющим фактором, т. к., только изменяя его, врач может влиять на исход операции. Однако в соответствии с рис. 2.24 степень его влияния надо признать весьма умеренной.

На рис. 2.25 показан фрагмент сравнения эмпирических и расчетных значений, позволяющий сделать вывод о необходимости повышения точности ИНС-модели, что будет сделано далее путем введения дополнительных нейронов в скрытый слой.

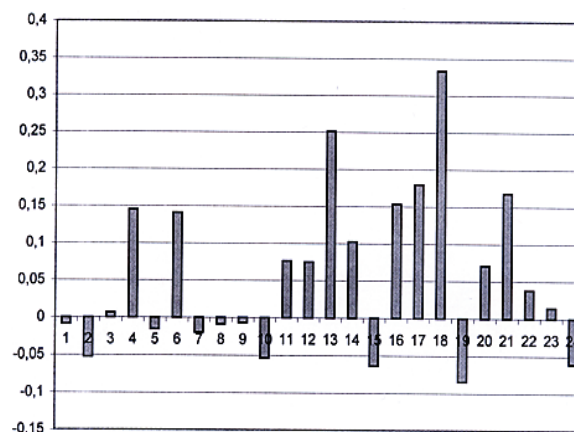


Рис. 2.24. Относительная чувствительность выхода ИНС-модели (исход) к входным параметрам: 1 – возраст больного; 2 – пол больного; 3 – нарушение мозгового кровообращения; 4 – почечная недостаточность; 5 – перенесенный острый инфаркт миокарда; 6 – сердечная недостаточность; 7 – дыхательная недостаточность; 8 – сахарный диабет; 9 – другие сопутствующие патологии; 10 – пульс; 11 – длительность заболевания; 12 – систологическое артериальное давление; 13 – частота дыхательных движений; 14 – лейкоциты крови; 15 – ядерный индекс интоксикации; 16 – наличие метастазов; 17 – размеры толстой кишки; 18 – размеры тонкой кишки; 19 – длительность операции; 20 – прорастание опухоли; 21 – наличие осложнений; 22 – рвота; 23 – наличие выпота в брюшной полости; 24 – объем операции

N17 (In)	N18 (In)	N19 (In)	N20 (In)	N21 (In)	N22 (In)	N23 (In)	N24 (In)	N26 (Out)	ModelVal
2	2	3	0	0	0	0	1	1	1,21828
2	1	3	0	0	0	0	1	1	0,90797
3	2	3	0	0	0	1	2	2	1,07926
2	1	1	0	0	0	0	2	1	0,87164
2	1	1	0	0	0	0	2	1	0,91313
2	2	2	0	0	0	1	2	1	1,07055
2	1	3	0	4	1	1	1	1	1,57656
2	2	1	0	0	1	0	2	1	1,23953
2	1	4	0	4	1	1	1	1	1,12754
3	3	3	0	4	1	1	2	2	1,66925
2	2	2	2	0	1	0	2	2	1,39074
3	2	2	0	0	0	1	2	2	1,64311
3	2	2	0	0	0	0	2	2	1,41487

Рис. 2.25. Фрагмент таблицы соответствия величин y , полученных из эмпирических данных (N26 (Out)) и по ИНС-модели (ModelVal) для однослойного персептрона – обведено красным

N15 (In)	N16 (In)	N17 (In)	N18 (In)	N19 (In)	N20 (In)	N21 (In)	N22 (In)	N23 (In)	N24 (In)	N26 (Out)	ModelVal
1	3	3	2	2	0	0	0	0	1	1	1,38282
3	2	3	2	2	0	0	0	1	1	1	1,36709
2	2	2	2	4	0	2	0	1	1	1	1,08387
2	3	3	1	2	2	0	0	1	2	2	1,80671
1	3	2	1	1	0	0	0	0	1	1	1,06511
2	1	3	1	2	0	1	1	1	2	1	1,29142
2	1	3	2	1	0	0	0	0	2	1	1,29654
1	3	2	1	1	0	0	0	0	1	1	1,16268
1	3	2	2	1	0	0	0	0	1	1	1,31938
3	3	3	3	2	2	4	1	1	2	2	2,07091
3	3	3	3	2	0	1	1	1	2	2	1,92575

Рис. 2.26. Фрагмент таблицы соответствия вличин y , полученных из эмпирических данных (N26 (Out)) и по ИНС-модели (ModelVal) для однослойного персептрона с линейным и квадратичным функциональными нейронами – обведено красным

N15 (In)	N16 (In)	N17 (In)	N18 (In)	N19 (In)	N20 (In)	N21 (In)	N22 (In)	N23 (In)	N24 (In)	N26 (Out)	ModelVal
3	3	2	1	3	2	0	1	0	1	1	1,22795
2	1	3	2	1	0	0	1	0	1	1	1,13719
1	3	3	2	2	0	0	0	0	1	1	1,39694
3	2	3	2	2	0	0	0	1	1	1	1,14885
2	2	2	2	4	0	2	0	1	1	1	0,9651
2	3	3	1	2	2	0	0	1	2	2	1,92817
1	3	2	1	1	0	0	0	0	1	1	0,96972
2	1	3	1	2	0	1	1	1	2	1	1,19631
2	1	3	2	1	0	0	0	0	2	1	0,94274
1	3	2	1	1	0	0	0	0	1	1	1,07349
1	3	2	2	1	0	0	0	0	1	1	1,10293
3	3	3	3	2	2	4	1	1	2	2	2,01006
3	3	3	3	2	0	1	1	1	2	2	1,99213

Рис. 2.27. Фрагмент таблицы соответствия вличин y , полученных из эмпирических данных (N26 (Out)) и по ИНС-модели (ModelVal) для однослойного персептрона с линейным, квадратичным и кубическим функциональными нейронами – обведено красным

Вычислительные эксперименты с ИНС-моделью, включающей линейный и квадратичный нейроны, в скрытом слое показали, что среднеквадратичная погрешность обучения составила 0,06, количество случаев неверного обучения – 7 из 156, что также нельзя считать приемлемым результатом.

В ходе вычислительных экспериментов пробовали также передаточную функцию нейрона типа «сигмоид», однако и в этом случае не были получены удовлетворительные результаты.

Удовлетворительное описание эмпирических данных получено лишь при введении в скрытый слой третьего – кубического нейрона (рис. 2.27). Этот результат свидетельствует о нелинейности исхода к входным факторам. Из рис. 2.27 видно также, что соответствие выхода модели и эмпирических данных находится на приемлемом уровне и после округления может быть использовано в ИНС-модели и экспертной системе.

Формальные результаты, полученные по данной модели: среднеквадратическая ошибка – 0,0324, число неправильных вычислений исхода – 2 на 156 случаев.

Необходимо отметить, что после анализа чувствительности входных параметров на исход была сделана попытка исключить слабо влияющие факторы из рассмотрения.

В ходе указанных действий из рассмотрения были удалены следующие факторы: 1 – возраст больного, 3 – нарушение мозгового кровообращения, 5 – перенесен-

ный острый инфаркт миокарда, 7 – дыхательная недостаточность, 8 – сахарный диабет, 9 – другие сопутствующие патологии, 23 – наличие выпота в брюшной полости.

Однако при их удалении погрешность моделей увеличилась. По этой причине было принято решение не удалять указанные связи.

Таким образом, в ходе предварительных исследований было показано, что наиболее адекватной структурой ИНС-модели является структура с одним линейным, одним квадратичным и одним кубическим нейронами в скрытом слое. Такая структура хорошо соответствует разложению функции многих переменных в ряд Тэйлора и показана на рис. 2.28.

При ее обучении с помощью разработанного комплекса программ получены следующие результаты.

ИНС-модель обучена на основе эмпирических данных. При этом получены значения весовых коэффициентов ИНС (рис. 2.28):

Mean square error of the ANN-model is: 2.8304260957E-02

The weight coefficients of the model are:

$w[1] = 2,3015998946E - 01$

$w[2] = 1,7209418639E + 00$

$w[3] = -1,2359808917E + 01$

$w[4] = 1,1115302471E + 01$

$w[5] = 2,7158935930E + 00$

w[6] = -1,1748413689E + 00
w[7] = -3,4919043683E + 00
w[8] = 2,0529339953E + 00
w[9] = -2,3700940981E + 00
w[10] = 1,0465730006E + 00
w[11] = 2,1188599892E - 01
w[12] = -1,4008000133E - 01
w[13] = -5,1521609583E - 01
w[14] = -2,1716430406E + 00
w[15] = 6,2039201483E - 01
w[16] = -1,1938880542E + 00
w[17] = 3,0240000415E - 02
w[18] = 1,8021000469E - 01
w[19] = 6,9544799760E - 01
w[20] = 3,9233101266E - 01
w[21] = -1,4157065810E + 00
w[22] = -2,0661396170E + 00
w[23] = 4,9599468701E + 00
w[24] = 7,2239003946E - 01
w[25] = 2,5577700011E - 02
w[26] = -2,4091679932E - 01
w[27] = 4,2861940457E - 01
w[28] = -7,7657501030E - 02
w[29] = -1,7685020290E - 01
w[30] = 1,5481900046E - 01
w[31] = -3,8774520120E - 01
w[32] = -3,5285900248E - 02
w[33] = -9,9368004084E - 03
w[34] = -8,2066999885E - 02
w[35] = -2,9356999954E - 02
w[36] = 1,5356579994E - 01
w[37] = 1,7336339902E - 01
w[38] = 5,4279399999E - 02
w[39] = -6,0626300059E - 02
w[40] = 1,8765500015E - 01
w[41] = -9,0538199600E - 02
w[42] = 2,1794060050E - 01
w[43] = -4,5073400097E - 02
w[44] = -6,7508899906E - 02
w[45] = -1,2132110007E - 01
w[46] = 2,3612400000E - 01
w[47] = 6,0452699949E - 02
w[48] = -1,8739859921E - 01
w[49] = -4,6936519690E - 01
w[50] = -3,6592360563E - 01
w[51] = 1,3161573742E + 00
w[52] = 9,1846354644E - 02
w[53] = -1,1995788002E + 00
w[54] = 1,6052586455E + 00
w[55] = 1,1784468166E + 00
w[56] = -1,6390204479E + 00
w[57] = 7,1836560487E - 01
w[58] = -2,1375539901E - 01
w[59] = 4,9231560747E - 01
w[60] = -7,0463719410E - 01
w[61] = -5,986478695E - 01
w[62] = 9,5872279907E - 01
w[63] = 2,0880299933E - 01
w[64] = 1,4178800334E - 01
w[65] = 7,1580381057E - 01
w[66] = 5,2050001310E - 01
w[67] = -4,2753861398E - 01
w[68] = -1,4150072994E - 02
w[69] = 2,7290770284E - 01

$$\begin{aligned}w[70] &= 3,1039979747E - 02 \\w[71] &= -2,6439161264E + 00 \\w[72] &= 3,0163000457E - 01 \\w[73] &= -5,3074314842E + 00 \\w[74] &= 3,0424280087E - 01 \\w[75] &= 3,7713756437E + 00 \\w[76] &= 4,6645099683E - 02 \\w[77] &= 1,3069249891E + 00 \\w[78] &= 3,1550467010E - 03 \\w[79] &= 4,9710360161E - 01\end{aligned}$$

Средняя квадратичная ошибка моделирования эмпирических данных без округления выхода модели составляет 0,028304260957, что при единице шкалы исхода в одну единицу (разница между 2 и 1) является хорошим результатом. Результаты представления эмпирических данных этой моделью с округлением выходных величин приведены ниже:

[illegible]

[illegible]

Значения коэффициентов w_1-w_{79} в модели (2.5) необходимо использовать с максимально возможной точностью, сохранив везде 11 значимых цифр в мантиссе числа.

973

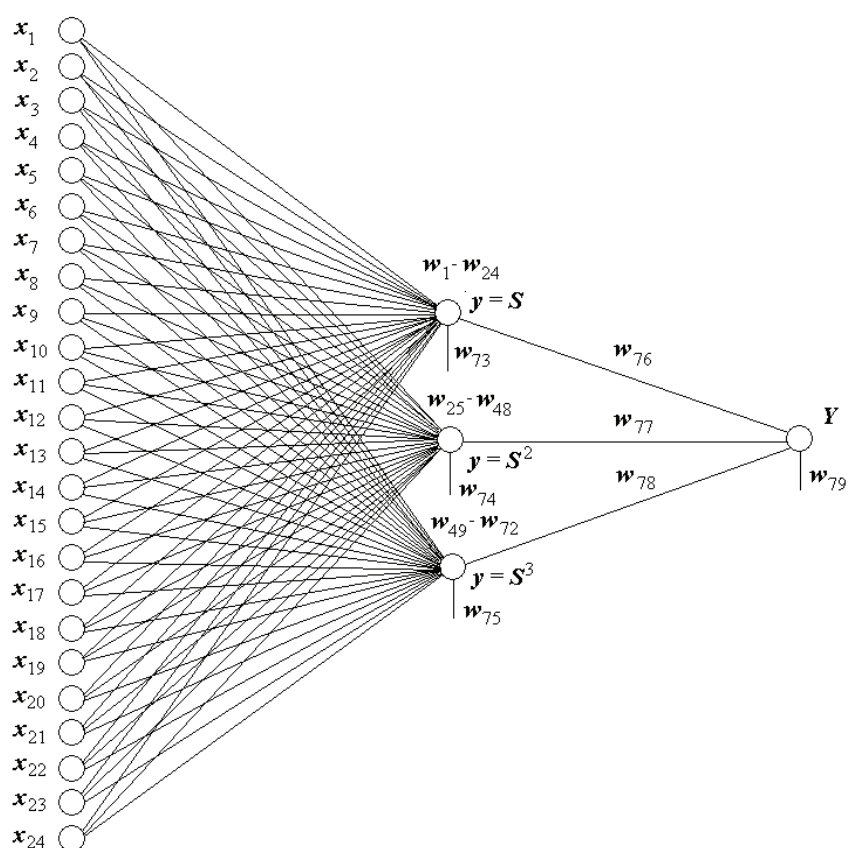


Рис. 2.28. Структурная разрабатываемой схема ИНС-модели

$$\begin{aligned}
 Y = & \left(w_{73} + \sum_{i=1}^{24} x_i w_i \right) w_{76} + \\
 & + \left[w_{74} + \left(\sum_{i=25}^{48} x_{i-24} w_i \right)^2 \right] w_{77} + \\
 & + \left[w_{75} + \left(\sum_{i=49}^{72} x_{i-48} w_i \right)^3 \right] w_{78} + w_{79}.
 \end{aligned} \quad (2.5)$$

Данное уравнение использовано для разработки компьютерной программы – экспертной системы, предназначенной для предоперационного исследования возможных исходов операции.

Экспертная система и алгоритм ее работы. На рисунке 2.29 представлена блок-схема алгоритма программы.

В блоках 1–3 происходит чтение эмпирических данных из файла ‘data’. Этот файл представляет собой прямоугольную матрицу, содержащую 156 строк (по числу пациентов) и 25 столбцов (24 входа и один выход-исход операции). Файл для удобства имеет тексто-

вый формат, что позволяет добавлять новую информацию при появлении сведений о новых больных.

В блоках 4–5 происходит чтение коэффициентов ИНС-модели из файла ‘coeff’. Этот файл представляет собой вектор-строку, содержащую 79 элементов. Файл для удобства имеет текстовый формат, что позволяет легко изменять коэффициенты модели при переучивании ИНС.

В блоке 6 пользователь вводит исходные данные по новому пациенту. При этом пользователю нет необходимости знать варианты кодировок. Он просто отвечает на предлагаемые вопросы и ставит соответствующие «галочки». После заполнения всех полей пользователь нажимает кнопку «Расчет исхода». При необходимости он может изменить один из показателей, например, сложность операции и снова нажать кнопку «Расчет исхода».

В блоках 7–9 производится расчет исхода по ИНС-модели (рис. 2.30 и уравнение (2.5)).

При необходимости результаты могут быть сохранены в текстовом файле (блок 10).

Таким образом, в ходе выполнения данного проекта (НИОКР) подтверждена возможность использования предлагаемых приемов, методов и программного обеспечения для разработки экспертных систем широкого назначения.

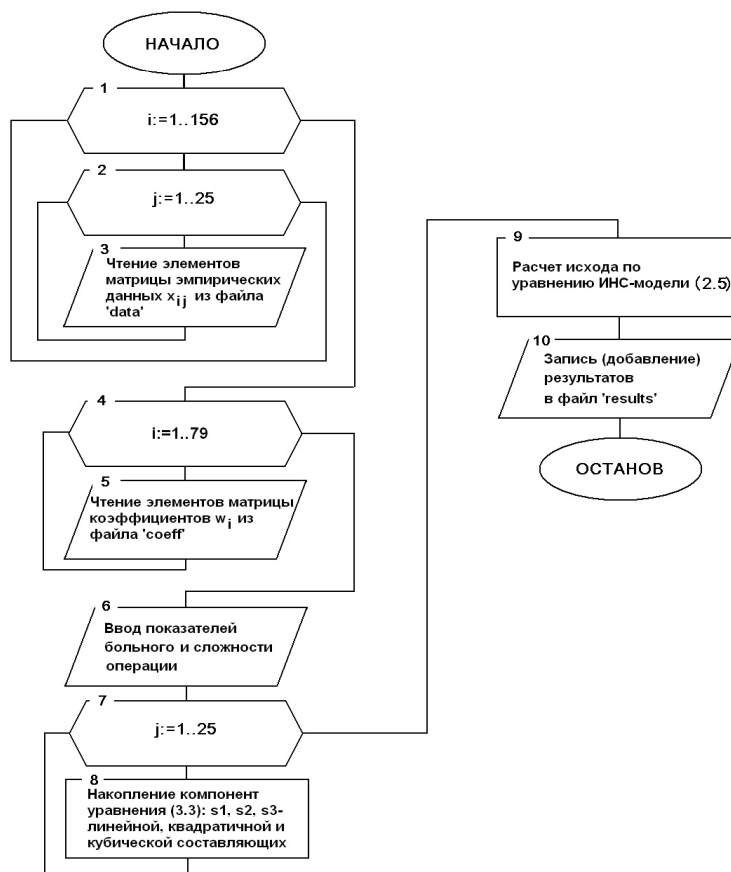


Рис. 2.29. Блок-схема алгоритма программы экспертной системы

Рис. 2.30. Интерфейс программы экспертной системы

2.4. Рекомендации к практическому использованию технологии и программно-технологического комплекса

Погрешности представления интеллектуальным ядром экспертной системы (ИНС-моделью) эмпирических данных являются приемлемыми для производственных и прочих целей. Указанное обстоятельство позволяет рекомендовать использование всей автоматизированной технологии разработки экспертных систем на основе ИНС-моделей с распределенным вводом информации и параллельными алгоритмами ее обработки, так и сами экспертные системы такого типа для решения производственных задач: прогнозирования, управления, оценивания, оптимизации и др.

Сравнение времени выполнения программы на единичном процессоре и вычислительном кластере показало, что существенная доля временных затрат получения ИНС-модели экспертной системы приходится на поиск структуры сети. Поэтому при имеющейся информации о реальной структуре объекта время разработки экспертной системы на основе ИНС-моделей может быть существенно сокращено путем исключения из процесса данной процедуры.

Применение указанной технологии и экспертных систем данного типа рекомендуется для внедрения (и коммерциализации) объектов, получение формализованных знаний для которых затруднено (производства, социальных процессов, медицины и др.).

ЗАКЛЮЧЕНИЕ

В ходе выполнения НИОКР по теме «Разработка автоматизированной технологии и программно-технологического комплекса для построения экспертных систем для различных сфер деятельности, включающей оригинальные способы и программные механизмы разработки экспертных систем, поддерживающих распределенный ввод данных, параллельные вычисления и имеющих интеллектуальное ядро, основанное на нейросетевых моделях» в ООО «Научно-производственная компания ЭКСПЕРТНЫЕ СИСТЕМЫ» по государственному контракту № 8894р/14330 от 11 апреля 2011 г. выявлены следующие основные положения.

1. Разработанная в ходе НИОКР автоматизированная технология и программно-технологический комплекс для построения экспертных систем для различных сфер деятельности показали работоспособность и применимость для различных типов объектов и могут быть рекомендованы для внедрения и коммерциализации.

2. Разработанные механизмы удаленного доступа позволяют исключить территориальную привязку экспертных систем, разрабатываемых по данной технологии и их обслуживание. Данное обстоятельство существенно повышает перспективы коммерциализации разработки ввиду того, что в перспективе такие ЭС могут быть размещены на сервере ООО «Научно-производственная компания ЭКСПЕРТНЫЕ СИСТЕМЫ», что существенно удешевляет их доработку, обслуживание и наладку.

3. Предложенные в ходе НИОКР аналитические модели позволяют оценивать и оптимизировать поэтапное обучение ИНС-моделей на основе поступаю-

щей от объекта информации, что позволяет разработать ЭС для широкого класса объектов. Такая широта также способствует успешной коммерциализации технологии и программно-технологического комплекса.

4. Разработанные механизмы параллельного обучения ИНС-моделей сокращают время их обучения на основе эмпирических данных и повышают конкурентную способность продукции на рынке экспертных систем.

5. Разработанное программное обеспечение подтверждено свидетельствами о регистрации программ для ЭВМ государственного образца. Право интеллектуальной собственности закреплено за ООО «Научно-производственная компания ЭКСПЕРТНЫЕ СИСТЕМЫ».

6. Результаты НИОКР опубликованы в печати [4–7, 61–69, 74–75]. По ним подготовлено 3 кандидатских диссертации и защищено 2 кандидатских диссертации.

7. По результатам НИОКР сделаны пробные разработки и продажи экспертных систем, что указывает на перспективность коммерциализации данной продукции.

ЛИТЕРАТУРА

1. Абдикеев Н.М. Проектирование интеллектуальных систем в экономике: учебник / под ред. Н.П. Тихомирова. М.: Экзамен, 2006. 528 с.
2. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. СПб.: Питер, 2004. 423 с.
3. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы: учебник. М.: Финансы и статистика, 2006. 424 с.
4. Арзамасцев А.А., Неудахин А.В., Зенкова Н.А. Автоматизированная технология построения экспертных систем с интеллектуальным ядром на основе ИНС-моделей // Открытое образование. М., 2008. Вып. 3 (68). С. 35–39.
5. Арзамасцев А.А., Троич К.Г., Зенкова Н.А., Неудахин А.В. Использование методов математического моделирования и искусственного интеллекта для оценки деятельности научных работников // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2008. Т. 13. Вып. 4. С. 301–312.
6. Арзамасцев А.А., Неудахин А.В. Методика разработки экспертных систем, использующих в качестве интеллектуального ядра ИНС-модели // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2008. Т. 13. Вып. 2–3. С. 219–222.
7. Арзамасцев А.А., Зенкова Н.А., Неудахин А.В. Технология построения медицинской экспертной системы на основе аппарата искусственных нейронных сетей // Информационные технологии. М., 2009. № 8. С. 60–63.
8. Гаврилов А.В. Системы искусственного интеллекта: учеб. пособие: в 2 ч. Новосибирск: Изд-во НГТУ, 2001. Ч. 1. 67 с.
9. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с.
10. Емельянов-Барковский Л.Б. Интеллектуальная квазибиологическая система. М.: Наука, 1990. 112 с.
11. Искусственный интеллект. Справочник: в 3 кн. / под ред. В.Н. Захарова, В.Ф. Хорошевского. М.: Радио и связь, 1990. Кн. 1. 426 с.; Кн. 2. 304 с.; Кн. 3. 368 с.
12. Козачков Л.В. Технология и инструментальные средства реализации информационно-экспертных систем // Сборник научных трудов. Львов: ВНИИМИУМ, 1989. С. 58–66.
13. Рыбина Г.В., Сикан К.В. Инструментальные средства поддержки разработки веб-ориентированных обучающих интегрированных экспертных систем // Интеллектуальные системы и технологии: сб. научных трудов научной сессии МИФИ-2008. М.: МИФИ, 2008. Т. 3. С. 45–49.
14. Рыбина Г.В. Интегрированные экспертные системы: современное состояние, проблемы и тенденции // Известия РАН. Теория и системы управления. 2002. № 5. С. 111–126.
15. Рыбина Г.В., Сикан К.В., Степанов Л.С. Методы и инструментальные средства разработки веб-ориентированных интегрированных экспертных систем // Программные продукты и системы. 2008. № 2. С. 25–28.

16. Рыбина Г.В., Сикан К.В., Курдюков А.Н. Принципы разработки веб-ориентированной версии комплекса АТ-ТЕХНОЛОГИЯ // Научная сессия МИФИ-2004: сб. научных трудов. М.: МИФИ, 2003. Т. 3. С. 62-63.
17. Сафонов В.О. Экспертные системы – интеллектуальные помощники специалистов. СПб.: Санкт-Петербургская организация общества «Знания» России, 1992.
18. Ясницкий Л.Н. Введение в искусственный интеллект. М.: Издательский центр «Академия», 2005. 176 с.
19. Duda R.O., Gaschnig J.G. Knowledge-based expert systems come of age // BYTE. 1981. V. 6. № 9. P. 238-281.
20. Harmon P. G2: Gensym's Real-Time Expert System // Intelligent Software Strategies. 1993. V. 9. № 3.
21. Kleschev A.S. Expert Systems Based on Metaknowledge: preprint. Vladivostok: Far Eastern Division of the Russian Academy of Sciences, 1994. 13 p.
22. Геловани В.А., Ковригин О.В. Экспертные системы в медицине. М.: Знание, 1987. 301 с.
23. Джексон П. Введение в экспертные системы. М.: Вильямс, 2001. 624 с.
24. Керридж А.Е. Использование экспертных систем // Нефть, газ и нефтехимия за рубежом. 1987. № 9. С. 107-110.
25. Козеренко Е.Б., Пузанов В.В. Сравнительный анализ оболочек экспертных систем // Новитех. 1990. № 1. С. 25-32.
26. Попов Э.В., Фоминых И.Б., Кисель Е.Б. Искусственный интеллект: справочник. Статические и динамические экспертные системы (классификация, состояние, тенденции). Методические материалы. М.: Центральный рос. дом знаний, 1995. 126 с.
27. Попов Э.В. Экспертные системы // Техническая кибернетика. 1987. № 5. С. 5-18.
28. Попов Э.В. Экспертные системы реального времени // Открытые системы. 1995. № 2 (10).
29. Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. М.: Наука, 1987. 252 с.
30. Сазыкин В.Г. Особенности решения задач экспертными системами реального масштаба времени // Приборы и системы управления. 1995. № 10. С. 11-14.
31. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. М.: Финансы и статистика, 1990.
32. Clements B.R., Preto F. Evaluating Commercial Real Time Expert System Software for Use in the Process Industries // C&I. 1993. P. 107-114.
33. Lee Ed.S., O'Keefe R.M. Developing a Strategy for Expert System Verification and Validation // IEEE Transactions on Systems, Man and Cybernetics. 1994. V. 24. № 4. P. 648-655.
34. Florentin J.J. Software Review: KEE // Expert Systems. 1987. V. 4. № 2. P. 18-120.
35. Hayes-Roth F., Jacobstein N. The State of Knowledge-Based Systems // Communications of the ACM. 1994. V. 37. № 3. P. 27-39.
36. Бельченко В.Е., Дедкова Т.Г., Частиков А.П. Инструментальные средства программирования экспертных систем: оболочки экспертных систем. Краснодар: КИТУ, 1994. 112 с.
37. Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование. М.: Вильямс, 2007. 1152 с.
38. Есеев О.В. Продукционная управляющая система для гибких производственных систем // Техническая кибернетика. 1987. № 5. С. 93-112.
39. Еришова О.В., Вальберг Н.Г., Чистякова Т.Б. Построение экспертной системы для управления карбидной печью // Математические методы в технике и технологиях: сб. 13 Международной научной конференции ММТТ-2000. СПб., 2000. Т. 6. С. 145-146.
40. Керов Л.А., Частиков А.П., Юдин Ю.В., Юхтенко В.А. Экспертные системы: инструментальные средства разработки: учеб. пособие / под ред. Ю.В. Юдина. СПб.: Политехника, 1996. 220 с.
41. Кирсанов Б.С., Попов Э.В. Состояние разработки инструментальных средств и экспертных систем // Искусственный интеллект. Справочник: в 3 кн. М.: Радио и связь, 1990. Кн. 1. С. 290-312.
42. Клецев А.С. Реализация экспертных систем на основе декларативных моделей представления знаний: препринт. Владивосток: ДВО АН СССР, 1988. 45 с.
43. Клецев А.С., Князева М.А. Экспертная система, моделирующая процесс оптимизации программ. Представление знаний: препринт. Владивосток: ИАПУ ДВО АН СССР, 1989. 25 с.
44. Компьютер обретает разум: пер. с англ. / под ред. В.Л. Стефанюка. М.: Мир, 1990.
45. Татжибаева О.А. Разработка экспертных систем: методические указания к расчетно-графическим работам по дисциплине «Системы искусственного интеллекта». Оренбург: ГОУ ОГУ, 2005. 23 с.
46. Уотерман Д. Руководство по экспертным системам: пер. с англ. М.: Мир, 1989. 388 с.
47. Behrouz H. Far, Koono Z. Ex-W-Pert System: Distributed Web-based Expert System for Groupware Design // Journal of Expert Systems With Applications. 1996. V. 11. № 4. P. 475-480.
48. Huntington D. Web-Based Expert Systems are on the Way: Java-based Web Delivery // PCAL. 2000. V. 14. № 6. P. 34-36.
49. Горбань А.Н., Душин-Барковский В.Л., Курдин А.Н., Миркес Е.М., Новоходько А.Ю., Россиев Д.А., Терехов С.А., Сенашова М.Ю., Царегородцев В.Г. Нейроинформатика. Новосибирск: Наука, Сибирское предприятие РАН, 1998. 296 с.
50. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996.
51. Горбань А.Н. Обучение нейронных сетей. М.: СП Параграф, 1990.
52. Каллан Р. Основные концепции нейронных сетей. М.: Вильямс, 2001. 288 с.
53. Осовский С. Нейронные сети для обработки информации / пер. с польск. И.Д. Рудинского. М.: Финансы и статистика, 2004. 344 с.
54. Хайкин С. Нейронные сети: полный курс: пер. с англ. 2-е изд. М.: Издательский дом «Вильямс», 2006. 1104 с.
55. Берштейн Л.С. Нечеткие модели для экспертных систем в САПР. М.: Энергоатомиздат, 1991. 136 с.
56. Голицын Г.А., Фоминых И.Б. Интеграция нейросетевой технологии с экспертными системами // Труды 5 национальной конференции по ИИ. Казань, 1996. С. 68-71.
57. Голицын Г.А. Применение нейросетевой технологии в ЭС // Экспертные системы реального времени: материалы семинара. М.: РДЗ, 1995. С. 55-60.
58. Потапова Т.Б. Структурная модель управления технологическим участком непрерывного производства как база знаний для экспертной системы // Приборы и системы управления. 1996. № 9. С. 27-29.
59. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного // Докл. АН СССР. 1957. Т. 114. № 5. С. 953-956 (цит. по: Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996).
60. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Докл. АН СССР. 1956. Т. 108. № 2. С. 179-182.
61. Арзамасцев А.А., Неудухин А.В. Автоматизированная технология построения экспертных информационных систем // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2008. Т. 13. Вып. 1. С. 83-85.
62. Арзамасцев А.А., Неудухин А.В. Концептуальные основы проектирования экспертных систем // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2009. Т. 14. Вып. 1: материалы 13 Всероссийской научной конференции преподавателей и аспирантов «Державинские чтения». С. 242-244.
63. Kryuchin O.V., Arzamastsev A.A., Troitzsch K.G. A parallel algorithm for selecting activation functions of an artificial neural network // Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 12. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_12_Arbeitsberichte.pdf. Загл. с экрана.
64. Kryuchin O.V., Arzamastsev A.A., Troitzsch K.G. A universal simulator based on artificial neural networks for computer clusters // Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 2. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_02_Arbeitsberichte.pdf. Загл. с экрана.
65. Kryuchin O.V., Arzamastsev A.A., Troitzsch K.G. Comparing the efficiency of serial and parallel algorithms for training artificial neural networks using computer clusters // Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 13. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_13_Arbeitsberichte.pdf. Загл. с экрана.
66. Kryuchin O.V., Arzamastsev A.A., Troitzsch K.G. The prediction of currency exchange rates using artificial neural networks // Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 4. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_04_Arbeitsberichte.pdf. Загл. с экрана.
67. Kryuchin O.V., Arzamastsev A.A., Troitzsch K.G., Zenkova N.A. Simulating social objects with an artificial neural network using a computer cluster // Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 15. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_15_Arbeitsberichte.pdf. Загл. с экрана.
68. Kryuchin O.V., Arzamastsev A.A., Zenkova N.A., Troitzsch K.G., Sletkov D.V. Simulating medical objects simulation using an artificial neural network whose structure is based on adaptive resonance theory //

- Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 14. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_14_Arbeitsberichte.pdf. Загл. с экрана.
69. *Zenkova N.A., Arzamastsev A.A., Troitzsch K.G.* Development of a technology of designing intelligent information systems for the estimation of social objects // Institut für Wirtschafts- und Verwaltungsinformatik Fachbereich Informatik Universität Koblenz-Landau. 2011. № 1. URL: http://www.uni-koblenz.de/~fb4reports/2011/2011_01_Arbeitsberichte. Загл. с экрана.
 70. *Арзамасцев А.А.* Математическое и компьютерное моделирование: учеб. пособие. Тамбов: Издательский дом ТГУ им. Г.Р. Державина, 2010. 257 с.
 71. *Арзамасцев А.А., Плотникова Ю.В.* Информационная система для оптимизации биотехнологического процесса на основе методов математического моделирования // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2008. Т. 13. Вып. 5. С. 376-394.
 72. *Арзамасцев А.А., Плотникова Ю.В.* Оптимизация биотехнологического процесса утилизации отходов производства этанола на основе методов математического моделирования // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2011. Т. 16. Вып. 1. С. 174-179.
 73. *Арзамасцев А.А., Неудакхин А.В., Зенкова Н.А., Сергеева М.С.* Разработка автоматизированной технологии построения экспертных информационных систем // Актуальные проблемы информатики и информационных технологий: сб. 12 Международной научно-практической конференции, Тамбов, 4-5 сент. 2008 г. Тамбов: ТГУ им. Г.Р. Державина, 2008. С. 115-117.
 74. *Неудакхин А.В., Арзамасцев А.А., Зенкова Н.А.* Построение нейросетевых экспертных систем с помощью автоматизированной технологии // Вопросы современной науки и практики. Университет им. В.И. Вернадского. Тамбов, 2009. № 4 (18). С. 179-183.
 75. *Arzamastsev A.A., Zenkova N.A., Troitzsch K.G., Neudakhin A.V.* Technology of intellectual information system design for estimation of social objects // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2011. Т. 16. Вып. 1. С. 169-173.

Поступила в редакцию 6 июня 2012 г.

Arzamastsev A.A., Zenkova N.A., Kryuchin O.V., Kvaschenkin D.O., Neudakhin A.V. AUTOMATED TECHNOLOGY AND SOFTWARE AND ENGINEERING COMPLEX FOR CONSTRUCTION OF EXPERT SYSTEMS WITH INTELLECTUAL CORE BASED ON NEURAL NETWORK MODELS, SUPPORT OF DISTRIBUTED DATA INPUT AND PARALLEL EQUATIONS

As a result of the project a set of software and technology, allowing you to create expert systems for various fields of activity is developed. Expert systems of this type are based on a distributed input data through a system of remote access, intellectual core, based on artificial neural networks with its training phase using parallel algorithms.

The software has been tested in the development of expert systems in the following areas: industrial object (expert system for process development of biomass based on waste of ethanol production), a medical facility (expert system to predict the outcome of surgical treatment of wasps, complications of colorectal cancer). With its use the models for the environmental objects (prediction shrimp population density in an open system and the air temperature in the city of Tambov), economic object (forecasting currency quotations), social object (model of professional predisposition state credit indebtedness, high school students) are also obtained. In all cases high quality match between empirical and calculated values due to the high adaptive capacity of the core of the expert system and procedures for a phased training is reached. The work identified: universalism proposed mechanisms of adaptation and expert systems subject to empirical data from a variety of subject areas, the possibility of practical use of these techniques and methods. The developed software package may have commercial use.

Key words: automated technology; expert systems; artificial neural network; intellectual core; parallel calculations; distributed data entry; remote access mechanisms.