

На правах рукописи

ОКУЛОВСКИЙ Юрий Сергеевич

ПРОГРАММНЫЙ КОМПЛЕКС ОБЕСПЕЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ
ВЫЧИСЛЕНИЙ

05.13.18. – математическое моделирование, численные методы и комплексы программ

А в т о р е ф е р а т
диссертации на соискание ученой степени
кандидата физико-математических наук

Екатеринбург – 2009

Работа выполнена на кафедре алгебры и дискретной математики ГОУ ВПО «Уральский государственный университет им. А.М. Горького»

Научный руководитель: доктор физико-математических наук, доцент
ПОПОВ Владимир Юрьевич

Официальные оппоненты: доктор физико-математических наук
БАЛАГАНСКИЙ Владимир Сергеевич

кандидат физико-математических наук
ВОЛКАНИН Леонид Сергеевич

Ведущая организация: ГОУ ВПО «Уральский государственный технический университет – УПИ им. первого президента России Б.Н. Ельцина»

Защита состоится октября 2009 года в часов на заседании диссертационного совета Д 212.286.10 по защите докторских и кандидатских диссертаций при ГОУ ВПО «Уральский государственный университет им. А.М. Горького» по адресу: 620000, г. Екатеринбург, пр. Ленина, 51, комн. 248

С диссертацией можно ознакомиться в Научной библиотеке ГОУ ВПО «Уральский государственный университет им. А.М. Горького».

Автореферат разослан сентября 2009 года

Ученый секретарь
диссертационного совета,
доктор физ.-мат. наук,
профессор

Пименов В.Г.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Под системами искусственного интеллекта обычно понимаются семейства различных алгоритмов решения задач, которые в процессе работы демонстрируют некоторое сходство с человеческой умственной деятельностью [19]. В частности, предполагается, что они способны делать выводы, являющиеся логическим следствием имеющихся знаний; обучаться решению задачи по предъявленным примерам решения и в дальнейшем работать на основании накопленного опыта; демонстрировать способность к адаптации и корректировать собственное поведение при изменении условий; находить решения задач, соответствующие указанным критериям правильного ответа без четко указанного алгоритма поиска решения. Системы искусственного интеллекта широко используются в различных областях. В частности, можно отметить такие области, как: поиск приближенных решений для труднорешаемых задач и NP-полных задач [5], [15]; моделирование экспертной оценки (в медицине [20], предсказании погоды [4], и т.д.); системное программное обеспечение (сжатие информации [23], распознавание вредоносного программного обеспечения [6]); робототехника (распознавание изображений и техническое зрение, планирование и управление [25]); экономика [12]; энергетика [21].

Интеллектуальные системы, как правило, запускаются на компьютерах традиционной архитектуры и с этой точки зрения являются обычными алгоритмами. В то же время, с практической точки зрения, даже при работе на обычной машине удобнее думать об интеллектуальных системах не как об алгоритмах, а как об альтернативных, не алгоритмических, способах решения задачи. Различие состоит в следующем. В традиционных, детерминированных алгоритмах последовательность действий строго фиксирована, и ее логика понятна человеку. В интеллектуальных алгоритмах последовательность действий не фиксирована. Чаще всего она не задана человеком, а возникает сама в результате обучения интеллектуальной системы. При этом человек лишь определяет рамки, в которых происходит обучение. Соответственно, логика обработки входных данных часто оказывается неизвестна программисту – система работает, но извне непонятно, как она это делает.

На сегодняшний день одним из основных методов анализа интеллектуальных систем является вычислительный эксперимент. В подавляющем большинстве случаев оказывается невозможным строго доказать корректность работы интеллектуальной системы или строго обосновать параметры системы, необходимые для корректного решения задачи. Возможно лишь попробовать решить задачу теми или иными конфигурациями интеллектуальной системы и убедиться в том, что задача решается. В связи с этим для исследований или применения интеллектуальных систем необходима их программная реализация. Непосредственная реализация интеллектуальных систем при решении практической задачи не является оправданной. Программирование интеллектуальных систем имеет свои отличительные особенности, требует специальных знаний и сложно в отладке.

В рамках настоящей работы под интеллектуальными системами понимаются преимущественно нейронные сети [21], [24], генетические алгоритмы [1], [22] и алгоритмы кол-

лективного разума [3], [5]. На сегодняшний день создано и постоянно создается большое количество программных реализаций систем искусственного интеллекта. Однако не существует известных систем, реализующих алгоритмы коллективного разума. Также не представлены продукты, интегрирующие генетические алгоритмы и нейронные сети. Широко распространены пакеты, содержащие лишь реализацию нейронных сетей. Успешные коммерческие реализации (см., например, [11], [13]) имеют закрытую архитектуру, и позволяют работать лишь со строго определенными типами нейронных сетей и алгоритмами их обучения, существенно ограничивая возможности пользователя по разработке новых систем. Большинство распространенных программных комплексов поддержки генетических алгоритмов (см., например, [7], [9]) не допускают интеграцию с нейронными сетями из-за узкой специализации. Также следует отметить более низкий уровень реализаций генетических алгоритмов по сравнению с нейронными сетями, проявляющийся в детальности документации, поддержке, спектре функциональности и т.д. Некоторые системы искусственного интеллекта реализованы в составе крупных программных пакетов, таких, как Mathlab ([10],[18]). Такая реализация позволяет полнее интегрировать интеллектуальные вычисления со сложными математическими вычислениями, однако затрудняет оптимизацию интеллектуальных систем и их использование вне среды Mathlab.

Исходя из перечисленных выше проблем с использованием готовых пакетов, формулируется основная цель работы.

Цель диссертации. Основная цель диссертации заключается в создании высокоэффективного программного комплекса, обеспечивающего высокое качество реализации интеллектуальных систем и удобство их встраивания в другое программное обеспечение. Для достижения этой цели, решаются три задачи:

- 1) теоретическое исследование моделей интеллектуальных систем для описания их вычислительных возможностей;
- 2) разработка универсальной архитектуры программного комплекса поддержки интеллектуальных вычислений, реализация в нем основных типов интеллектуальных вычислений, а также сопутствующих утилит;
- 3) тестирование комплекса на стандартных задачах и путем решения прикладных задач.

Методы исследования. Основными методами теоретических исследований являются методы дискретной математики, геометрии, математической логики и теории алгоритмов. В прикладных исследованиях используются методы теории нейронных сетей и теории эволюционных систем, а также методы объектно-ориентированного программирования, автоматного программирования и порождающего программирования.

Научная новизна. Основные теоретические результаты, полученные в данной работе, являются новыми. Разработан программный комплекс, который впервые интегрирует

три основных типа интеллектуальных вычислений, обеспечивая комплексный подход к интеллектуальным вычислениям.

Теоретическая и практическая ценность. Работа содержит теоретическую и практическую части. Теоретические результаты работы могут быть использованы для выбора оптимальной структуры нейронных сетей при анализе векторных множеств. Результаты прикладных исследований представляют самостоятельный научный интерес. Созданный программный комплекс предоставляет гибкую архитектуру и широкие возможности для интеграции, за счет которых возникает самостоятельное направление реализации дополнительных типов интеллектуальных вычислений в рамках комплекса. Программный комплекс также представляет ценность как инструмент для проведения исследований при решении задач из областей компьютерных наук, физики, медицины и т.д.

Апробация работы. Основные результаты работы докладывались на следующих конференциях: Международная алгебраическая конференция (Екатеринбург, 2005), 37-я региональная молодежная конференция в Кунгурке (Кунгурка, 2006), 9-я международная конференция «Интеллектуальные системы и компьютерные науки» (Москва, 2006), Международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» (Екатеринбург, 2006), 38-я региональная молодежная конференция в Кунгурке (Кунгурка, 2007), Четвертая сибирская школа-семинар по параллельным и высокопроизводительным вычислениям (Томск, 2007), Международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» (Екатеринбург, 2007), IV Межвузовская научно-практическая конференция студентов, аспирантов и молодых ученых «Безопасность информационного пространства» (Тюмень, 2007), Научная сессия МИФИ-2008 (Москва, 2008), 39-й Всероссийская молодежная конференция в Кунгурке (Кунгурка, 2008), XXIII международная конференция «Уравнения состояния вещества» (Эльбрус, 2008), Третья международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» (Екатеринбург, 2008), 40-я Всероссийская молодежная конференция в Кунгурке (Кунгурка, 2009), VI Всероссийская межвузовская конференция молодых ученых (Санкт-Петербург, 2009).

Результаты работы докладывались и обсуждались на семинарах «Алгебраические системы», «Системы искусственного интеллекта» и «Системном семинаре» УрГУ, семинаре по теории функций в Институте математики и механики УрО РАН. Устройства под управлением интеллектуальных систем, реализованных с помощью программного комплекса, демонстрировались на выставках: «Образовательная среда-2008» (г. Москва, сентябрь 2008); IV Уральская венчурная выставка-ярмарка «Инновации-2008» (г. Екатеринбург, ноябрь, 2008); Отчетная выставка по Приоритетному Национальному Проекту Образование (г. Москва, декабрь 2008); «Планета Education-2009» (г. Екатеринбург, февраль, 2009).

Публикации. Основные результаты диссертации изложены в [26]–[48]. Работы [26], [27], [35], [36], [39]–[42], [44], [48] написаны в неразрывном соавторстве с другими авторами. В работе [43] диссертантом проведена разработка и реализация интеллектуального алгоритма обработки данных и численные эксперименты. В работе [28] диссертантом выполнены формулировка и доказательство всех результатов, связанных с линейно-порожденными частичными порядками.

Работы [26]–[28] опубликованы в изданиях, входивших в перечень ВАК на момент публикации.

Структура и объем работы. Работа состоит из 111 страниц машинописного текста, включающего введение, три главы и библиографический список из 41 наименования.

Краткое содержание работы.

В первой главе проведены теоретические исследования точных вычислительных возможностей отдельного нейрона. Данные исследования продолжают исследования других авторов (см., например, [14]) в области вычислительных возможностей нейронных сетей. Выбранное направление исследования является фундаментальным для таких важнейших и актуальных на сегодняшний день проблем, как описание минимальных вычислительных возможностей нейронной сети заданной конфигурации или выбора оптимальной нейронной сети, выполняющей заданную функцию.

Исследуются функции $f : \mathbb{L} \rightarrow \mathbb{R}$, где $\mathbb{L} \subset \mathbb{R}^m$, и $\mathbb{L}$ не более чем счетно. Такая функция может быть представлена одним нейроном, если $f(X) \equiv g(A \cdot X + a_0)$, где A – вектор весов, $a_0 \in \mathbb{R}$, а g – функция активации нейрона, определяющая его тип. Рассматривались модели персептрона, т.е. нейрона с функцией активации $g(x) = \text{sign}(x)$, а также n -порогового нейрона, для которого g является кусочно-постоянной функцией с n разрывами. Показано, что один n -пороговый нейрон способен представить любую функцию на конечном носителе $\mathbb{L}$.

Исследованы линейно-разделимые булевы функции, т.е. булевы функции, которые могут быть представлены одним персептроном, и построена их частичная классификация. Найден индуктивный критерий линейной разделимости с помощью оператора конкатенации. Если $f_1 = f_1(x_1, \dots, x_{n-1})$, $f_2 = f_2(x_1, \dots, x_{n-1})$ и

$$f(x_1, \dots, x_{n-1}, x_n) = (\neg x_n \wedge f_1(x_1, \dots, x_{n-1})) \vee (x_n \wedge f_2(x_1, \dots, x_{n-1})),$$

то f является конкатенацией f_1 и f_2 , $f = f_1 * f_2$.

Теорема 1. Пусть $\mathcal{H}(f)$ – множество всех векторов A , для которых существует a_0 такое, что

$$f(X) \equiv \text{sign}(A \cdot X + a_0).$$

Тогда $f_1 * f_2$ линейно разделима, если и только если $\mathcal{H}(f_1) \cap \mathcal{H}(f_2)$ имеет внутреннюю точку.

На основании этого критерия сформулирован алгоритм порождения списка линейно-разделимых функций (ЛРФ) арности n из списка ЛРФ арности $n - 1$, перебирающий все пары функций арности $n - 1$ и проверяющий, что конкатенация этой пары является ЛРФ.

Обнаружено, что множество ЛРФ имеет сложную внутреннюю структуру, использование которой позволяет существенно сократить алгоритм порождения. Любая ЛРФ может быть приведена к монотонной ЛРФ путем применения отрицаний к ее аргументам. Установлено, что монотонная ЛРФ всегда есть конкатенация монотонных ЛРФ, что позволило ускорить алгоритм за счет поиска только монотонных ЛРФ.

Далее, выделен класс канонических ЛРФ, определяемых следующим образом. Введем понятие ранга i -ого аргумента:

$$r_i = |\{(x_1, \dots, x_n) : x_i = 1 \wedge f(x_1, \dots, x_n) = 1\}|.$$

Булеву функцию, аргументы которой упорядочены по возрастанию ранга, назовем канонической. Доказано, что любая монотонная функция приводится к канонической путем перестановки аргументов, и что любая каноническая ЛРФ есть конкатенация канонических ЛРФ. Последний факт позволяет дальнейшую оптимизацию алгоритма порождения.

Сужение классов ЛРФ можно продолжить. На основании такого сужения построена частичная классификация ЛРФ. Основной идеей является построение последовательности классов $\mathfrak{N}_k(n)$, где $0 \leq k \leq n$. Любой представитель класса $\mathfrak{N}_k(n)$ является конкатенацией представителей $\mathfrak{N}_k(n-1)$, и, следовательно, алгоритм может быть распространен на любой класс. Вводятся специальные функционалы, которые позволяют перевести функцию из $\mathfrak{N}_k(n)$ в $\mathfrak{N}_{k+1}(n)$.

Сформулируем полученные результаты строго. Введем некоторые необходимые определения. Рассмотрим n -мерный вектор $X = (x_1, \dots, x_n)$ и индексное множество $M = \{m_1, \dots, m_k\}$, $1 \leq m_i \leq n$. Проекцией X на M назовем вектор $X|_M = (x_{m_1}, \dots, x_{m_k})$. Дополнительным индексным множеством $\bar{M}$ называется такое множество, для которого $M \uplus \bar{M} = \{1, \dots, n\}$. Наконец, если $|M| = k$, $Y \in \mathbb{R}^k$, $Z \in \mathbb{R}^{(n-k)}$, $V \in \mathbb{R}^n$, $V|_M = Y$, $V|_{\bar{M}} = Z$, то V называется смещением Y и Z по M и обозначается как $X \circ_M Y$.

Пусть E_k является k -мерным вектором вида $(1, -1, \dots, -1)$, а O_k является k -мерным вектором вида $(-1, 1, \dots, 1)$. Назовем булеву функцию нерегулярной на M , если существуют Y_1, Y_2 такие, что $f(E_k \circ_M Y_1) > f(O_k \circ_M Y_1)$ и $f(E_k \circ_M Y_2) < f(O_k \circ_M Y_2)$. В противном случае f регулярна на M . Функция f является регулярной, если она регулярна на любом индексном множестве. Регулярная функция f является инвариантной на M , если для всех X выполняется $f(E_k \circ_M X) = f(O_k \circ_M X)$. Наконец, регулярная и неинвариантная функция является верхней на M , если для некоторого X выполняется $f(E_k \circ_M X) > f(O_k \circ_M X)$, и нижней в противном случае. Регулярные функции и линейно-разделимые функции связывает следующая теорема:

Теорема 2. *Всякая линейно-разделимая функция регулярна на любом индексном множестве M .*

Обозначим через $\mathfrak{N}_k(n)$ множество всех линейно-разделимых функций, которые являются верхними либо инвариантными относительно всех M таких, что $|M| \leq k$. Справедливо, что $\mathfrak{N}_0(n)$ – множество всех линейно-разделимых функций арности n , $\mathfrak{N}_1(n)$ – множество всех монотонных ЛРФ, а $\mathfrak{N}_2(n)$ – множество всех канонических ЛРФ. Для любого k возможна дальнейшая оптимизация алгоритма: если $f_1 * f_2 \in \mathfrak{N}_k(n)$, то $f_1, f_2 \in \mathfrak{N}_k(n-1)$.

Для перевода функции из одного класса в другой введены ξ и ζ функционалы, обобщающие применение к аргументу отрицания и перестановку аргументов:

$$[\xi, M]f(X \circ_M Y) = \begin{cases} f(X \circ_M Y) & , \quad X \neq E_k, X \neq O_k \\ f(E_k \circ_M Y) & , \quad X = O_k \\ f(O_k \circ_M Y) & , \quad X = E_k \end{cases},$$

$$[\zeta, M]f(X \circ_M Y) = \begin{cases} [\xi, M]f, & \text{если } f \text{ нижняя на } M \\ f, & \text{иначе} \end{cases}.$$

Показано, что с помощью конечного числа применений ζ -функционалов возможно приведение функции к особому классу ζ -замкнутых. Известными примерами ζ -замкнутых функций являются функции, таблица истинности которых заполнена сначала нулями, а затем – единицами. Такие функции мы называем примитивными. Для важного случая функций, сводимого к примитивным, построена классификация булевых функций, схематично изображенная на следующей диаграмме.

$$\begin{array}{ccccccc} \mathfrak{N}_1(1) & \xrightarrow{*} & \mathfrak{N}_1(2) & \xrightarrow{*} & \mathfrak{N}_1(3) & \xrightarrow{*} & \mathfrak{N}_1(4) \xrightarrow{*} \dots \\ & & \downarrow \zeta & & \downarrow \zeta & & \downarrow \zeta \\ & & \mathfrak{N}_2(2) & \xrightarrow{*} & \mathfrak{N}_2(3) & \xrightarrow{*} & \mathfrak{N}_2(4) \xrightarrow{*} \dots \\ & & & & \downarrow \zeta & & \downarrow \zeta \\ & & & & \mathfrak{N}_3(3) & \xrightarrow{*} & \mathfrak{N}_3(4) \xrightarrow{*} \dots \\ & & & & & & \downarrow \zeta \\ & & & & & & \dots \end{array}$$

Стрелки, помеченные $*$, указывают на то, что один класс может быть получен из другого путем применения конкатенации, а стрелки, помеченные ζ – что функции первого класса сводятся к другому ζ -преобразованиями. Класс $\mathfrak{N}_k(n)$ является подмножеством класса $\mathfrak{N}_k(n)$, содержащим те функции, которые могут быть сведены к примитивным. Если класс ζ -замкнутых функций не исчерпывается примитивными функциями, это означает лишь появление новых «предельных точек» в классификации.

Исследование вычислительных способностей n -порогового нейрона проведено с использованием упорядочений конечного множества векторов по значению скалярного произведения с данным вектором. Введено понятие линейно-порожденного порядка. Порядком в нашей работе называется транзитивное и рефлексивное отношение. Рассмотрим некоторое множество векторов $\mathcal{A} \subset \mathbb{R}^n$. Линейно-порожденный порядок (ЛПП), порожденный $\mathcal{A}$ — это порядок ρ такой, что $X\rho Y$ тогда и только тогда, когда для всех $A \in \mathcal{A}$ выполняется

$A \cdot X \leq A \cdot Y$. Множество всех векторов, порождающих порядок ρ , обозначим через $\text{Cone}(\rho)$. Простейшим примером линейно-порожденного порядка является линейно-порожденный линейный порядок. Такие порядки могут быть наведены на множестве одним вектором A , для которого $A \cdot X \neq A \cdot Y$ для всех $X, Y \in \mathbb{L}$. Множество линейно-порожденных линейных порядков обозначается как $\mathfrak{L}_1$.

На линейно-порожденных порядках введены операции инфимума и супремума. Порядок τ называется инфимумом порядков ρ и σ , если $\text{Cone}(\tau) = \text{Cone}(\rho) \cap \text{Cone}(\sigma)$. Инфимум обозначается как $\rho \wedge \sigma$. Минимальное множество, содержащее $\mathfrak{L}_1$ и замкнутое относительно $\wedge$, обозначим как $\mathfrak{L}_2$. Супремумом двух порядков ρ и σ будем называть порядок $\tau = \rho \vee \sigma$, образующее множество которого является наименьшим возможным объединением конусов из $\mathfrak{L}_2$, содержащим $\text{Cone}(\rho)$ и $\text{Cone}(\sigma)$, которое при этом является замкнутым коническим множеством. Минимальное множество, содержащее $\mathfrak{L}_1$ и замкнутое относительно $\vee$ и $\wedge$ обозначим через $\mathfrak{L}$. Доказано, что $\mathfrak{L}$ содержит все линейно-порожденные порядки и только их, и что оно является конечной решеткой относительно операций супремума и инфимума. Исследованы некоторые свойства этой решетки. В частности, показано, что решетка ЛПП на любой проекции носителя $\mathbb{L}$ изоморфно вложима в решетку ЛПП исходного множества. То же самое справедливо для решеток ЛПП подмножеств $\mathbb{L}$.

Указана связь между линейно-порожденными порядками и функциями, вычислимыми с помощью одного n -порогового нейрона. Порядок ρ будем называть k -дольным, если существует разбиение $\mathcal{D} = \{D_1, \dots, D_k\}$ такое что для любых $i < j$ и $X \in D_i, Y \in D_j$ выполняется $X \rho Y$.

Теорема 3. *Функция $f : \mathbb{L} \rightarrow \{0, \dots, k\}$ может быть вычислена k -пороговым нейроном тогда и только тогда, когда существует невырожденный, линейно-порожденный, $(k+1)$ -дольный порядок ρ такой, что доли ρ в точности равны $f^{-1}(0), \dots, f^{-1}(k)$. Запись $f^{-1}(i)$ обозначает полный прообраз i .*

Во второй главе описан программный комплекс поддержки интеллектуальных вычислений GANS¹. Указаны основные отличия комплекса от существующих аналогов. Комплекс поддерживает все основные типы интеллектуальных вычислений в рамках единой архитектуры. В настоящее время не существует других программных комплексов с указанным свойством. Поддержка разных типов интеллектуальных вычислений в рамках одной архитектуры позволяет использовать один и тот же подход при решении всех задач, требующих интеллектуальных вычислений. Кроме того, она позволяет строить эффективные комбинированные системы, состоящие из, например, взаимодействующих нейронных сетей и генетических алгоритмов. Комплекс одинаково удобно может использоваться в образовании, научных исследованиях и написании прикладных программ. Удобство использования комплекса как платформы для изучения интеллектуальных вычислений происходит из его

¹Полная документация, исходный код и примеры доступны по адресу <http://ai.math.usu.ru/wiki/doku.php/нир/gans>

блочной архитектуры, которая позволяет осваивать комплекс постепенно. Для начального уровня достаточно лишь научиться составлять интеллектуальную систему из готовых блоков, а для более тонкой настройки возможно написание новых компонент в рамках имеющейся архитектуры. В научных исследованиях важна открытость архитектуры комплекса, позволяющая использовать новые алгоритмы вычислений совместно с имеющимися. Прикладное значение комплекса продиктовано из его реализации на современной платформе программирования .NET [16], [17], в виде библиотеки. Такая реализация допускает обширные средства интеграции интеллектуальных вычислений в приложения. Кроме того, выбранная платформа реализации позволяет запускать комплекс под разными операционными системами, и писать дополнительный код для комплекса на разных языках программирования.

Нами разработана блочная архитектура, которая позволяет собирать сложные системы из простых компонентов. Данная архитектура позволила представить пакет GANS в виде набора сравнительно простых компонентов, соединение которых позволяет построить сложную интеллектуальную систему. Доводка системы осуществляется путем замены одних компонентов на другие, а также изменением параметров компонентов. Компоненты обеспечивают автоматическое сохранение системы на диск и ее восстановление. Это позволило разработать движок обучения, проводящий обучение в течение длительного времени с автосохранением, ведением журнала и восстановлением работы системы после ошибок, отключения питания и перезагрузки операционной системы. Таким образом, разработчику и пользователю пакета нет необходимости самостоятельно реализовывать сервисные задачи. Кроме того, блочная архитектура существенно снижает объем кодирования – программный код должен лишь описать сбор компонент интеллектуальной системы.

Важным результатом является приложение компонентной модели к сборке нейронных сетей и алгоритмов их обучения. Нами также разработана декомпозиция нейронных сетей и обучающих алгоритмов на простые блоки, которая позволила представлять сложные сети и алгоритмы обучения, описанные в литературе, в виде композиций простых. В результате, обеспечение многообразия алгоритмов обучения и архитектур сетей, которые можно наблюдать, например, в [21] и [24], на практике сводится лишь к сборке их из блоков. Основной сущностью для сборки нейронных сетей является компонент нейронной сети. Компонент нейронной сети представляет собой некую подсеть с заданным количеством входов и выходов. Элементарным компонентом является слой нейронов одного типа, где все нейроны имеют одинаковое число входов. Компоненты могут связываться различными способами. Так, при последовательном соединении выход первого компонента передается следующему. При рекуррентном соединении выход компонента перенаправляется на его же вход. Такие композиции позволяют получить из элементарных слоев самые разные архитектуры нейронных сетей: многослойный персептрон, сеть прямого распространения сигнала, составленную из разных типов нейронов, а также рекуррентные сети (например, сети Эльмана или RTRN [21]).

Аналогичное решение применено для описания алгоритмов обучения нейронных сетей. Алгоритм обучения нейронной сети вырабатывает направление изменения весов нейрон-

ной сети. В комплексе GANS введены уточняющие алгоритмы, которые не вырабатывают совершенно новое направление, а уточняют направление предыдущего в цепочке алгоритма. Были разработаны различные способы комбинаций алгоритмов, и показано, что многие описанные в литературе сложные алгоритмы могут быть легко разложены в комбинацию простых.

Помимо реализации нейронных сетей, в проекте GANS реализованы также генетические алгоритмы и алгоритмы коллективного разума. Данные типы интеллектуальных вычислений обычно трудны для реализации, поскольку допускают существенную вариативность: так, в генетических алгоритмах генетический код может быть не только массивом, но и, в общем случае, любой структурой данных, с произвольными операциями скрещивания и мутации. В алгоритмах коллективного разума поведение объектов может быть, в общем случае, произвольным алгоритмом. Нами выявлены основные принципы работы этих алгоритмов, реализован шаблон вычислений, не ограничивающий общность, и сделаны стандартные реализации для широко распространенных подходов: ген в виде массива с набором необходимых операций, а также реализация муравьиных алгоритмов и искусственных иммунных систем в рамках единого шаблона алгоритма коллективного разума.

Также исследован вопрос о языке описания интеллектуальных систем. В качестве основы для такого языка использован язык текстовых преобразований Thorn, также разработанный автором диссертации². Документ Thorn состоит из команд, вложенных одна в другую. В отличие от HTML, синтаксис Thorn существенно компактнее, поэтому документы Thorn удобно набирать с клавиатуры. Основное различие между языком Thorn и другими распространенными языками разметки обеспечивается компилятором. Каждая команда ассоциируется с некоторой программой, которая может быть написана на любом языке программирования (как правило, используется Perl). При разборе текста компилятор собирает все текстовые данные, связанные с командой. После этого, он запускает ассоциированную программу, передавая ей текстовые данные как входные параметры. Программа возвращает строку, которая подставляется на место команды. После того как все команды выполнены, трансляция документа завершается. Программы могут обмениваться между собой данными через глобальные переменные. Программы, ассоциированные с командами, сохраняются в библиотеках, которые отделены от документа Thorn. Один и тот же документ Thorn может быть транслирован разными библиотеками, т.е. с различным значением команд. Например, различные библиотеки могут производить код для разных языков программирования.

Язык Thorn успешно применяется для создания HTML-файлов, где он удобен за счет малого, по сравнению с HTML, объема разметочной информации. Кроме того, возможно применение языка Thorn для порождения документов на языке программирования (например, C#). Подобное решение, называемое кодогенерацией, лежит в русле совре-

²Спецификация языка Thorn доступна по адресу
<http://ai.math.usu.ru/wiki/doku.php/нир/thornado-2/главная>

менного подхода к разработке программного обеспечения, который говорит о том, что программное обеспечение должно порождаться спецификацией. В этом смысле документ Thorn является спецификацией программного обеспечения, а сгенерированный код – собственно, программным обеспечением, заведомо удовлетворяющим спецификации. Набор библиотек для генерации кода фактически описывают некий малый язык для кодогенерации некоторой части приложения. Нами разработана концепция Thornado построения малых языков на основе Thorn для генерации исходного кода программ, успешно примененная для создания языков описания пользовательского интерфейса, печатных форм документов и др. В рамках диссертации описана эта концепция и создание на ее основе описания интеллектуальных систем на основе GANS.

В третьей главе рассматривается тестирование созданного комплекса на различных примерах. Для отладки интеллектуальных систем, для сравнения показателей быстродействия, а также для использования пакета в учебно-методической работе, в комплекс GANS встроены готовые примеры. Эти примеры разбиты на три основных класса: решение NP-трудных задач, обработка изображений и аппроксимация функций. Тестовые задачи являются типовыми, т.е. большинство проблем, традиционно решаемых интеллектуальными вычислениями, так или иначе к ним сводятся.

Также пакет GANS был использован для решения ряда практически востребованных задач. Одним из традиционных применений нейронных сетей является распознавание изображений. Целью распознавания изображений является логическая характеристика участков изображения: какой предмет изображен на этом участке, каким свойством обладает изображенный предмет в данном участке, и т.д. Распознавание изображений как часть технического зрения является одним из основных направлений развития робототехники. Нами были решены две проблемы распознавания изображений с использованием нейронных сетей: распознавание результатов ультразвукового исследования сердечной мышцы, выполненное совместно с Институтом иммунологии и физиологии РАН [44], и построение трасс движений игроков на футбольном поле [38].

Второй рассмотренной задачей является калибровка параметров физической модели для соответствия экспериментальным данным на примере решения важной задачи современной теплофизики. Свойства жидкостей обычно изучаются в стабильных состояниях. Жидкости могут пребывать в стабильных состояниях так долго, как это требуется, и, следовательно, допускают повторяющиеся измерения своих параметров. Однако, при нестабильных состояниях такие измерения затруднительны, и свойства жидкостей в нестабильных состояниях часто оказываются недостаточно изученны. Изучались свойства жидкостей при их быстром нагревании. Физическая суть эксперимента состоит в измерении зависимости температуры жидкости от времени, при нагревании жидкости тонким зондом. Целью математической составляющей эксперимента является подбор параметров жидкости и зонда таким образом, чтобы данные, полученные вычислениями по математической модели, совпадали с данными, полученными в результате экспериментальных измерений.

Таким образом, по опосредованным данным в виде зависимости температуры от времени, устанавливаются свойства жидкостей. Эта задача успешно решена с помощью генетических алгоритмов. Свойства жидкости и другие параметры эксперимента кодируются в виде массива чисел с плавающей точкой. Затем алгоритм подбирает оптимальные значения, стараясь минимизировать отклонение расчетной зависимости от экспериментальной [43]. Разработанное программное обеспечение внедрено в исследовательский процесс Института теплофизики УрО РАН.

Другим применением генетических алгоритмов стало решение задачи калибровки робототехнических устройств. Традиционно система управления робототехническим устройством первоначально реализуется на некоторой модели, а потом переносится непосредственно на робота. При этом необходим процесс калибровки, который позволяет настроить параметры модели таким образом, чтобы изменения в модели соответствовали реальному отклику устройства. Параметры могут быть получены непосредственными измерениями, однако такой подход часто недостаточно точен, или требует введения в конструкцию дорогостоящего измерительного оборудования. Мы приводим алгоритм, который автоматизирует калибровку по обратной связи, работает в условиях шума, и потенциально может адаптироваться к изменениям параметров устройства. При этом мы расширяем понятие калибровки, включая в список параметров все физические данные о роботе и его окружении. Фактически, приведенный алгоритм позволяет роботу самостоятельно изучать окружающую среду и выполнять в ней целевые задачи, притом что первоначально информация о среде представлена в роботе лишь в виде модели взаимодействия, безо всяких численных значений. Данная задача успешно решена с помощью двух взаимодействующих генетических алгоритмов. Первый алгоритм пытается объяснить имеющуюся экспериментальную базу, подбирая наборы калибровочных параметров, описывающие соответствие условий эксперимента его результату. Второй анализирует возможные варианты калибровочных параметров и подбирает эксперимент, который бы подтвердил одни наборы и опроверг другие. Таким образом, каждый из алгоритмов использует результаты другого для вычисления собственной целевой функции. Установлено, что такая комбинация алгоритмов может весьма быстро калибровать робототехническую руку [47].

Результаты диссертации.

1. Создан универсальный программный комплекс поддержки интеллектуальных вычислений, в нем реализованы все основные интеллектуальные системы: нейронные сети, генетические алгоритмы и алгоритмы коллективного разума.
2. Создано вспомогательное программное обеспечение для работы интеллектуальных алгоритмов и создания их из описания на языке разметки.
3. Построена классификация линейно-разделимых булевых функций путем сведения их к простому ядру с применением специальных функционалов.
4. Описана структура упорядочений конечных множеств по значению их скалярного

произведения с весами нейронов.

5. Построены алгоритмы распознавания объектов на видеоизображениях и изображениях ультразвукового исследования.

6. Построены алгоритмы автоматической калибровки робототехнических устройств по обратной связи.

7. Построен алгоритм калибровки параметров физической модели при исследовании перегретых жидкостей.

Список литературы

- [1] Baricelli N. A. Symbiogenetic evolution processes realized by artificial methods. *Methodos*. – 1957. V. 9. N 35-36. PP. 143-182.
- [2] Cover T. M. The Number of Linearly Inducible Orderings of Points in d-Space. *SIAM Journal on Applied Math.* – 1967. V. 15. N 2. PP. 434-439.
- [3] Dasgupta D., Atttoh-Okine N. Immunity-Based Systems: A survey. *IEEE International Conference on Systems, Man and Cybernetics*. – 1997. PP. 369-374.
- [4] Diyankov O. V., Lykov V. A., Terekhoff S. A. Artificial neural networks in weather forecasting. *RNNS/IEEE Symposium on Neuroinformatics and Neurocomputers*. – 1992.
- [5] Dorigo M., Stutzle T. *Ant Colony Optimization*. Cambridge: MIT Press – 2004.
- [6] Forrest S., Perelson A. S., Allen L., Cherukuri R. Self-Nonself discrimination in a Computer. *IEEE Symposium on Research in Security and Privacy*. – 1994. PP. 202-212.
- [7] GALib Documentation. Massachusetts Institute of Technology. [Электронный ресурс]. URL: <http://lancet.mit.edu/galib-2.4/> (дата обращения 19. 08. 2009).
- [8] Harrington J. *Code generation in action*. Greenwich: Manning – 2003.
- [9] Jenes. Genetic algorithms for Java. Computational and Intelligent Systems Engineering Laboratory at University of Sannio. [Электронный ресурс]. URL: <http://sites.google.com/a/ciselab.org/jenes/Home/> (дата обращения 21. 07. 2009).
- [10] Neural Network Toolbox 6. 0. 2. The MathWorks. [Электронный ресурс]. URL: <http://www.mathworks.com/products/neuralnet/> (дата обращения 11. 08. 2009).
- [11] *Neurosolutions User's Manual*. Gainesville: NeuroDimension Inc. – 1995.
- [12] Paulo J G Lisboa, Vellido A., Edisbury B. *Business Applications of Neural Networks: The State-Of-The-Art of Real-World Applications (Progress in Neural Processing)*. New Jersey: World Scientific Publishing Company – 2002.
- [13] Peltarion Synapse Documentation. Peltarion Inc. [Электронный ресурс]. URL: <http://www.peltarion.com/webdoc/> (дата обращения 16. 08. 2009).
- [14] Sima J., Orponen P. *A Computational Taxonomy and Survey of Neural Network Models. of Numbers and Symbols. (BS 1749:1985)* London: British Standards Institution. – 2001. PP. 2965–2989.
- [15] Spears W. M. *Using Neural Networks And Genetic Algorithms As Heuristics For NP-Complete Problems*. Fairfax Country, Virginia: George Mason University– 1989.
- [16] Thai T. L., Lam H. *NET Framework Essentials. Second Edition*. Cambridge: O'Reilly – 2002.
- [17] The C# language. Microsoft. [Электронный ресурс]. URL: <http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx> (дата обращения 13. 08. 2009).

- [18] The Genetic Algorithm Optimization Toolbox (GAOT) for Matlab 5. North Carolina State University. [Электронный ресурс]. URL: <http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/> (дата обращения 19. 08. 2009).
- [19] Turing A. Computing Machinery and Intelligence. Mind. – 1950. V. LIX. N 236. PP. 433-460.
- [20] Ежов А. А., Чечеткин В. Д. Нейронные сети в медицине. Откр. сист. – 1997. № 4. Стр. 34-37.
- [21] Осовский С. Нейронные сети для обработки информации. Москва: Финансы и статистика – 2002.
- [22] Рутковская Д., Рутковский Л., Пильинский М. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва: Горячая линия-Телеком – 2004.
- [23] Семенюк В. В. Экономное кодирование. Санкт-Петербург: СПбГИТМО – 2001.
- [24] Хайкин С. Нейронные сети: полный курс. Второе издание. Москва: Вильямс – 2006.
- [25] Юревич Е. И., Макаров И. М., Каляев И. А., Лохин В. М. Интеллектуальные роботы. Москва: Машиностроение – 2007.

Статьи автора по теме диссертации, опубликованные в ведущих рецензируемых научных журналах, определенных ВАК:

- [26] Деев Д. В., Окуловский Ю. С., Попов В. Ю., Часовских В. П. Система кодогенерации Thornado и ее использование для создания бизнес-приложений. Научно-технический вестник СПбГУ ИТМО. №57. Санкт-Петербург: СПбГУИТМО – 2008. Стр. 80-87.
- [27] Окуловский Ю. С., Шека А. С. Об архитектуре роботов и интеллектуальном управлении ими. Научно-технический вестник СПбГУ ИТМО. №48. Санкт-Петербург: СПбГУИТМО – 2008. Стр. 143-150.
- [28] Окуловский Ю. С., Попов В. Ю. О сложности распознавания множества векторов одним нейроном. Математические труды. Т. 12, №1. Новосибирск: Издательство Института Математики им. С. Л. Соболева СО РАН – 2009. Стр. 130-143.

Прочие статьи автора по теме диссертации:

- [29] Окуловский Ю. С. Язык ТН. Проблемы теоретической и прикладной математики: Труды 37-й Всероссийской молодежной конференции. Екатеринбург: УрО РАН – 2006. Стр. 496-500.

- [30] Окуловский Ю. С. A model and implementation of universal engine for neural systems. 9-й международная конференция «Интеллектуальные системы и компьютерные науки» труды конференции, II том. Москва: Изд-во механико-математического факультета МГУ – 2006. Стр. 21-24.
- [31] Окуловский Ю. С. О функциях, реализуемых одним нейроном. Международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» Тезисы докладов. Екатеринбург: УГТУ-УПИ – 2006. Стр. 241-241.
- [32] Окуловский Ю. С. Алгоритмы порождения линейно-разделимых функций. Проблемы теоретической и прикладной математики: Труды 38-й Всероссийской молодежной конференции. Екатеринбург: УрО РАН – 2007. Стр. 382-387.
- [33] Окуловский Ю. С. О функциях, реализуемых одним нейроном. Проблемы математического моделирования и информационно-аналитической поддержки принятия решений. Выпуск 3. Екатеринбург: УГТУ-УПИ – 2007. Стр. 55-61.
- [34] Морнев М. Л., Окуловский Ю. С. Комплекс распределения интеллектуальных вычислений на кластер под управлением операционной системы Windows. Четвертая сибирская школа-семинар по параллельным и высокопроизводительным вычислениям. Программа и тезисы. Томск – 2007. Стр. 49-50.
- [35] Морнев М. Л., Окуловский Ю. С., Попов В. Ю., Шека А. С. Проблемы интеллектуального моделирования колесных роботов. Международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» Тезисы докладов. Екатеринбург: УГТУ-УПИ – 2007. Стр. 229-231.
- [36] Деев Д. В., Окуловский Ю. С. Разработка бизнес-приложений с использованием системы кодогенерации Thornado. Международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» Тезисы докладов. Екатеринбург: УГТУ-УПИ – 2007. Стр. 208-210.
- [37] Окуловский Ю. С. О классификации линейно-разделимых функций. 10-ая Всероссийская научно-техническая конференция «Нейроинформатика-2008, сборник научных трудов, часть 1. Москва: Изд-во МИФИ – 2008. Стр. 212-218.
- [38] Окуловский Ю. С. Распознавание фигур на игровом поле. Труды конференции «Научная сессия МИФИ-2080». Москва: Изд-во МИФИ – 2008. Стр. 193-194.
- [39] Бабинов А. М., Морнев М. Л., Окуловский Ю. С., Попов В. Ю., Шека А. С. Базовое программное обеспечение колесного робота на базе RC-CAR AT-10ES. Труды конференции «Научная сессия МИФИ-2080». Москва: Изд-во МИФИ – 2008. Стр. 221-222.
- [40] Деев Д. В., Окуловский Ю. С. О порождении подсистемы вывода документов. Проблемы теоретической и прикладной математики: Труды 39-й Всероссийской молодежной конференции. Екатеринбург: УрО РАН – 2008. Стр. 388-392.

- [41] Бабинов А. М., Морнев М. Л., Окуловский Ю. С., Попов В. Ю., Шека А. С. Об интеллектуальных алгоритмах управления роботами. Прикладные аспекты моделирования и разработки систем информационно-аналитической поддержки и принятия решений. Выпуск 4. Екатеринбург: УГТУ-УПИ – 2008. Стр. 169-175.
- [42] Деев Д. В., Окуловский Ю. С. Система кодогенерации Thornado. Прикладные аспекты моделирования и разработки систем информационно-аналитической поддержки и принятия решений. Выпуск 4. Екатеринбург: УГТУ-УПИ – 2008. Стр. 209-216.
- [43] Скрипов П. В., Смотрицкий П. В., Старостин А. А., Окуловский Ю. С. Estimation of thermophysical properties for pulse heated substances. Experiment and modelling. XXIII международная конференция «Уравнения состояния вещества» тезисы докладов. Эльбрус – 2008. Стр. 66-68.
- [44] Окуловский Ю. С., Попов В. Ю. О построении контуров на изображениях УЗИ и МРТ. Методы компьютерной диагностики в биологии и медицине – 2008. Саратов – 2008. Стр. 82-83.
- [45] Окуловский Ю. С., Попов В. Ю. Компендиум. Результаты реализации инновационных образовательных программ ПНПО в части формирования актуальных компетенций и развития межвузовского взаимодействия в области информационно-коммуникационных технологий к Всероссийскому семинару-презентации. Нижний Новгород – 2008. Стр. 135.
- [46] Окуловский Ю. С. Интеллектуальные алгоритмы калибровки робототехнических устройств. Третья международная научная конференция «Информационно-математические технологии в экономике, технике и образовании» Тезисы докладов. Екатеринбург: УГТУ-УПИ – 2008. Стр. 244.
- [47] Окуловский Ю. С. Интеллектуальный алгоритм автоматической калибровки робототехнических устройств. Проблемы теоретической и прикладной математики: Труды 40-й Всероссийской молодежной конференции. Екатеринбург: УрО РАН – 2009. Стр. 359-363.
- [48] Конончук Д. О., Окуловский Ю. С. Универсальный пакет поддержки интеллектуальных вычислений GANS. VI Всероссийская межвузовская конференция молодых ученых. Труды конференции. Выпуск 4. Математическое моделирование и программное обеспечение. Санкт-Петербург: СПбГУИТМО – 2009. Стр. 151-157.