

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО АТОМНОЙ ЭНЕРГИИ
РОССИЙСКАЯ АКАДЕМИЯ НАУК
РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ
МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

НАУЧНАЯ СЕССИЯ МИФИ–2005

НЕЙРОИНФОРМАТИКА–2005

**VII ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

**ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ**

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2005

УДК 001 (06)+004.032.26 (06)

ББК 72я5+32.818я5

М82

НАУЧНАЯ СЕССИЯ МИФИ–2005. VII ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2005»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. – М.: МИФИ, 2005. – 214 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 26–28 января 2005 года в МИФИ в рамках VII Всероссийской конференции «Нейроинформатика–2005».

Материалы лекций связаны с рядом проблем, актуальных для современного этапа развития нейроинформатики, включая ее взаимодействие с другими научно-техническими областями.

Ответственный редактор

Ю. В. Тюменцев, кандидат технических наук

ISBN 5–7262–0556–1

© *Московский инженерно-физический институт
(государственный университет), 2005*

Содержание

Предисловие	7
<i>А. И. Самарин. Нейронные сети с преднастройкой</i>	13
Введение	14
Нейронные сети У. Мак-Каллока	17
Общая постановка задачи	22
Неоднозначность интерпретации сенсорной информации в условиях замыкания обратной связи через окружение	25
Нейронные сети с преднастройкой функциональных преобразований	26
Нейросетевые механизмы константности восприятия при формировании сенсомоторной координации робота	36
Заключение	38
Литература	40
<i>В. А. Головкин. Нейросетевые методы обработки хаотических процессов</i>	43
Введение	44
Методы анализа хаотических процессов	45
Основные определения	45
Анализ временного ряда	48
Выбор временной задержки	49
Выбор размерности пространства вложения	52
Определение старшего показателя Ляпунова	57
Понятие старшего показателя Ляпунова	57
Алгоритм Бенеттина	59
Оценка старшего показателя Ляпунова по выборке из одной наблюдаемой координаты	60
Оценка старшего показателя Ляпунова по выборке из одной наблюдаемой координаты с помощью нейронной сети	62
Определение спектра Ляпунова	64
Понятие спектра Ляпунова	65
Численный расчет спектра Ляпунова	66
 УДК 001 (06)+004.032.26 (06) Нейронные сети	 3

Нейросетевой метод оценки спектра Ляпунова на основе полной выборки	68
Нейросетевой метод оценки спектра Ляпунова по выборке из одной координаты	74
Результаты экспериментов	74
Прогнозирование и реконструкция хаотических процессов	76
Прогнозирование хаотических временных рядов	76
Реконструкция аттракторов	81
Общая методика обработки хаотических процессов	83
Заключение	86
Литература	88
С. А. Терехов. Адаптивные нейросетевые методы в многошаговых играх с неполной информацией	92
Введение: От задач оптимизации и оптимального управления к теории игр	93
Специфика теоретико-игровых постановок информационных задач	96
Стратегические игры в матричной форме	97
Равновесие в играх с полной информацией	98
Равновесия в смешанных стратегиях	101
Многошаговые игры	103
Модель многошаговой игры с неполной информацией	107
Задача о деятельности холдинга консалтинговых компаний	112
Модель бизнес-процессов	112
Формализация математической модели многошаговой игры	115
Индивидуальные стратегии игроков	117
Компьютерная система COGITO	121
Выборочные результаты моделирования	122
Обсуждение	122
Послесловие	126
Благодарности	128
Литература	128
Приложение. Нейросеть CNLS (Connectionist Normalized Local Splines)	130
Задачи	132
Задача 1. Ошибка в формуле	132
Задача 2. Плотность распределения континуума оптималь- ных решений в игре со стохастической матрицей	134

L. Rutkowski. Neuro-fuzzy inference systems	136
Introduction	137
Non-flexible neuro-fuzzy systems	137
Description of fuzzy systems	137
Mamdani-type neuro-fuzzy systems	141
Logical-type neuro-fuzzy systems	141
Generalized neuro-fuzzy systems	142
Flexibility in fuzzy systems	142
Compromise operators	142
Weighted triangular norms	143
Parameterized triangular norms	144
Soft fuzzy norms	145
Flexible neuro-fuzzy systems	146
Compromise neuro-fuzzy systems	152
Learning procedures	153
Simulation results	154
Final remarks: Design of flexible neuro-fuzzy systems	163
References	163
Н. Г. Ярушкіна. Нечеткие нейронные сети в когнитивном моделировании и традиционных задачах искусственного интеллекта	166
Введение	167
Нечеткие нейронные сети (ННС)	170
Определение нечеткого множества	170
Определение нечеткой системы	170
Системы нечеткого вывода	171
Определения И- и ИЛИ-нейронов	172
Определение нечеткой нейронной сети (ННС)	174
ANFIS — адаптивная НС, основанная на системе нечеткого вывода	174
Задачи когнитивного моделирования: Возможности ННС	176
Цели и задачи когнитивной науки	176
Задача моделирования человеческого восприятия	177
Синергетический компьютер	177
Задача исследования баланса право- и левополушарных методов	178
ННС — сочетание образного восприятия и логической обработки	178

Задача самоорганизации знаний	180
Нечеткая кластеризация	180
Задача исследования соотношения осознанного и подсознательного	182
ННС и возможности генерации правил поведения	182
Искусственный интеллект, нейроинтеллект, естественный интеллект	183
Задачи искусственного интеллекта: Возможности ННС	184
Научное направление «искусственный интеллект» (ИИ)	184
Состояние ИИ	185
Искусственный и вычислительный интеллект	186
Традиционная задача ИИ – инженерия знаний и экспертные системы	187
Мягкие экспертные системы	188
Пример мягкой ЭС для проектирования радиоэлектронных изделий	190
Задачи прикладного ИИ: Возможности ННС	193
Интеллектуальный анализ данных	193
ННС как Data Miner для нечетких реляционных баз данных	193
Извлечение правил из реляционных кортежей с помощью ННС	199
Нейросетевое решение задачи прикладного ИИ – развитие интеллектуальных информационных хранилищ	206
Определение информационного хранилища (ИХ)	206
Основные функции системы управления ИХ	206
Нейросетевая система управления ИХ	208
Нечеткая кластеризация информационных ресурсов с помощью сети Кохонена. Морфологический анализ текста. Стемминг	209
Классификация информационного ресурса на основе ННС	210
Заключение	212
Литература	212

ПРЕДИСЛОВИЕ

1. В этой книге содержатся тексты лекций, прочитанных на Пятой Школе-семинаре «Современные проблемы нейроинформатики», проходившей 26–28 января 2005 года в МИФИ в рамках VII Всероссийской научно-технической конференции «Нейроинформатика–2005».

При отборе и подготовке материалов для лекций авторы и редактор следовали принципам и подходам, сложившимся при проведении четырех предыдущих Школ (см. [1–7]). А именно, основной целью Школы было рассказать слушателям о современном состоянии и перспективах развития важнейших направлений в теории и практике нейроинформатики, о ее применениях. При этом особенно приветствовались лекции междисциплинарные, лежащие по охватываемой тематике «на стыке наук», рассказывающие о проблемах не только собственно нейроинформатики (т. е. о проблемах, связанных с нейронными сетями, как естественными, так и искусственными), но и о взаимосвязях нейроинформатики с другими областями мягких вычислений (нечеткие системы, генетические и другие эволюционные алгоритмы и т. п.), с системами, основанными на знаниях, с традиционными разделами математики, биологии, психологии, инженерной теории и практики.

Основной задачей лекторов, приглашаемых из числа ведущих специалистов в области нейроинформатики и ее приложений, смежных областей науки, было дать живую картину современного состояния исследований и разработок, обрисовать перспективы развития нейроинформатики в ее взаимодействии с другими областями науки.

2. В программу Школы-семинара «Современные проблемы нейроинформатики» на конференции «Нейроинформатика–2005» вошли следующие пять лекций:

1. *А. И. Самарин*. Нейронные сети с преднастройкой.
2. *В. А. Головки*. Нейросетевые методы обработки хаотических процессов.
3. *С. А. Терехов*. Адаптивные нейросетевые методы в многошаговых играх с неполной информацией.
4. *Л. Рутковский*. Нейро-нечеткие системы вывода.
5. *Н. Г. Ярушкина*. Нечеткие нейронные сети в когнитивном моделировании и традиционных задачах искусственного интеллекта.

Основные темы, рассматриваемые в этих лекциях, связаны с исследованиями междисциплинарного характера, в которых помимо обязательной

в лекциях Школы нейросетевой составляющей, используются также подходы и методы других научно-технических областей: теория игр (лекция *С. А. Терехова*), нечеткая логика (лекции *Л. Рутковского* и *Н. Г. Ярушкиной*), нелинейная динамика (лекция *В. А. Головки*).

3. В лекции **А. И. Самарина** «Нейронные сети с преднастройкой» рассматривается задача динамического предпрограммирования нейронных сетей при реализации сложных функциональных зависимостей, заданных в неявном виде. Показано, что введение свойств локальной адаптации позволяет существенно уменьшить число итерационных циклов в процессе поиска искомой зависимости, а распределенное представление функции позволяет заменить процедуру обучения процедурой преднастройки сети комбинаторным вытормаживанием вставочных нейронов. Описывается использование такой сети с преднастройкой при решении задач идентификации объектов управления.

4. Лекция **В. А. Головки** «Нейросетевые методы обработки хаотических процессов» примыкает по тематике затрагиваемых вопросов к лекциям *Н. Г. Макаренко* «Фракталы, аттракторы, нейронные сети и все такое» на Школе 2002 года [11] и «Эмбедология и нейропрогноз» на Школе 2003 года [12]. Ее предмет – нейросетевые методы обработки хаотических процессов. Излагаются такие вопросы, как определение параметров вложения динамической системы и идентификации хаоса. Рассмотрены нейросетевые методы вычисления наибольшего показателя Ляпунова и спектра Ляпунова. Исследовано применение нейронных сетей для построения странных аттракторов на основе прогнозируемых значений хаотического процесса. В качестве исходных данных использовались последовательности Энона и Лоренца. Показано, что при помощи нейронной сети можно осуществить реконструкцию динамики системы, используя малый объем исходных данных. Рассмотрен метод переобучения нейронной сети для увеличения горизонта прогнозирования хаотических процессов.

9. Лекция **С. А. Терехова** «Адаптивные нейросетевые методы в многошаговых играх с неполной информацией» в определенном смысле продолжает тематику его же лекций на Школах 2002 года [8], 2003 года [9] и 2004 года [10]. Она содержит изложение одного из подходов к управлению сложными динамическими системами. Этот подход основан на математической теории игр. Одна из важных задач, изучаемых теорией игр – это игры нескольких игроков с неполной информацией. Для этого случая формирование игроком оптимальной смешанной стратегии принятия решений

предлагается осуществлять с помощью вероятностного алгоритма обучения. Вводимые принципиальные положения иллюстрируются подробным практическим примером теоретико-игрового моделирования в оптимизации совместной бизнес-деятельности группы компаний, предоставляющих консалтинговые услуги. Основой рассматриваемой модели является нейросетевая аппроксимации функции цены состояния игроков. При этом используется нейронная сеть специального вида — локальные нормализованные сплайны, которая позволяет эффективно решать рассматриваемую задачу.

5. В лекции **Л. Рутковского** «Нейро-нечеткие системы вывода» рассматривается новый класс нейро-нечетких систем вывода, основанных на логическом подходе и на подходе, предложенном Мамдани. Рассматриваются средства обеспечения гибкости при проектировании нейро-нечетких систем. Для предлагаемого подхода характерно автоматическое формирование правил нечеткого вывода в процессе обучения системы. Высокая точность нейро-нечетких систем демонстрируется на примере задачи моделирования динамического объекта.

6. Лекция **Н. Г. Ярушкиной** «Нечеткие нейронные сети в когнитивном моделировании и традиционных задачах искусственного интеллекта» продолжает развитие темы, начатой автором в лекции на Школе 2004 года [13]. Она посвящена рассмотрению нечетких нейронных сетей (ННС), которые реализуют системы нечеткого логического вывода нейросетевыми методами. Возможности ННС иллюстрируются на широком спектре задач, включая соотношение логического и образного в мышлении, осознанного и неосознанного, обобщенного и конкретного. В лекции сопоставлены задачи инженерии знаний традиционного искусственного интеллекта с возможностями нейросетевых технологий. Приведены примеры использования ННС для создания прикладных интеллектуальных систем.

* * *

Для того, чтобы продолжить изучение вопросов, затронутых в лекциях, можно порекомендовать такой уникальный источник научных и научно-технических публикаций, как цифровая библиотека **ResearchIndex** (ее называют также **CiteSeer**, см. позицию [14] в списке литературы в конце предисловия). Эта библиотека, созданная и развиваемая отделением фирмы **NEC** в США, содержит уже несколько миллионов публикаций, причем это число постоянно и быстро увеличивается за счет круглосуточной работы поисковой машины.

Каждый из хранимых источников (статьи, препринты, отчеты, диссертации и т.п.) доступен в полном объеме в нескольких форматах (PDF, PostScript и др.) и сопровождается очень подробным библиографическим описанием, включающим, помимо данных традиционного характера (авторы, заглавие, место публикации и/или хранения и др.), также и большое число ссылок-ассоциаций, позволяющих перейти из текущего библиографического описания к другим публикациям, «похожим» по теме на текущую просматриваемую работу. Это обстоятельство, в сочетании с весьма эффективным полнотекстовым поиском в базе документов по сформулированному пользователем поисковому запросу, делает библиотеку ResearchIndex незаменимым средством подбора материалов по требуемой теме.

Помимо библиотеки ResearchIndex, можно рекомендовать также богатый электронный архив публикаций [15], недавно открывшийся поисковый сервис Google Scholar [16], а также портал научных вычислений [17].

Перечень проблем нейроинформатики и смежных с ней областей, требующих привлечения внимания специалистов из нейросетевого и родственных с ним сообществ, далеко не исчерпывается, конечно, вопросами, рассмотренными в предлагаемом сборнике, а также в сборниках [1–7].

В дальнейшем предполагается расширение данного списка за счет рассмотрения насущных проблем собственно нейроинформатики, проблем «пограничного» характера, особенно относящихся к взаимодействию нейросетевой парадигмы с другими парадигмами, развиваемыми в рамках концепции мягких вычислений, проблем использования методов и средств нейроинформатики для решения различных классов прикладных задач. Не будут забыты и взаимодействия нейроинформатики с такими важнейшими ее «соседями», как нейробиология, нелинейная динамика, численный анализ и т.п.

Замечания, пожелания и предложения по содержанию и форме лекций, перечню рассматриваемых тем и т.п. просьба направлять электронной почтой по адресу tium@mai.ru Тюменцеву Юрию Владимировичу.

Литература

1. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // III Всероссийская научно-техническая конференция «Нейроинформатика-2001», 23–26 января 2001 г. / Отв. ред. Ю. В. Тюменцев. – М.: Изд-во МИФИ, 2001. – 212 с.

2. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // IV Всероссийская научно-техническая конференция «Нейроинформатика-2002», 23–25 января 2002 г. / Отв. ред. Ю. В. Тюменцев. Часть 1. – М.: Изд-во МИФИ, 2002. – 164 с.
3. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // IV Всероссийская научно-техническая конференция «Нейроинформатика-2002», 23–25 января 2002 г. / Отв. ред. Ю. В. Тюменцев. Часть 2. – М.: Изд-во МИФИ, 2002. – 172 с.
4. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // V Всероссийская научно-техническая конференция «Нейроинформатика-2003», 29–31 января 2003 г. / Отв. ред. Ю. В. Тюменцев. Часть 1. – М.: Изд-во МИФИ, 2003. – 188 с.
5. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // V Всероссийская научно-техническая конференция «Нейроинформатика-2003», 29–31 января 2003 г. / Отв. ред. Ю. В. Тюменцев. Часть 2. – М.: Изд-во МИФИ, 2003. – 180 с.
6. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // VI Всероссийская научно-техническая конференция «Нейроинформатика-2004», 28–30 января 2004 г. / Отв. ред. Ю. В. Тюменцев. Часть 1. – М.: Изд-во МИФИ, 2004. – 200 с.
7. Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики» // VI Всероссийская научно-техническая конференция «Нейроинформатика-2004», 28–30 января 2004 г. / Отв. ред. Ю. В. Тюменцев. Часть 2. – М.: Изд-во МИФИ, 2004. – 200 с.
8. Терехов С. А. Нейросетевые аппроксимации плотности распределения вероятности в задачах информационного моделирования // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2002. – с. 94–120.
9. Терехов С. А. Введение в байесовы сети // В сб.: «Лекции по нейроинформатике». Часть 1. – М.: Изд-во МИФИ, 2003. – с. 149–187.
10. Терехов С. А. Нейро-динамическое программирование автономных агентов // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2003. – с. 111–139.
11. Макаренко Н. Г. Фракталы, аттракторы, нейронные сети и все такое // В сб.: «Лекции по нейроинформатике». Часть 2. – М.: Изд-во МИФИ, 2002. – с. 121–169.
12. Макаренко Н. Г. Эмбедология и нейропрогноз // В сб.: «Лекции по нейроинформатике». Часть 1. – М.: Изд-во МИФИ, 2003. – с. 86–148.

13. Ярушкіна Н. Г. Нечеткие нейронные сети с генетической настройкой // В сб.: «Лекции по нейроинформатике». Часть 1. – М.: Изд-во МИФИ, 2004. – с. 151–198.
14. NEC Research Institute CiteSeer (also known as ResearchIndex) – Scientific Literature Digital Library.
URL: <http://citeseer.nj.nec.com/>
15. The Archive arXiv.org e-Print archive – Physics, Mathematics, Nonlinear Sciences, Computer Science.
URL: <http://arxiv.org/>
16. Google Scholar.
URL: <http://scholar.google.com/>
17. Портал научных вычислений (Matlab, Fortran, C++ и т. п.)
URL: <http://www.mathtools.net/>

Редактор материалов выпуска,
кандидат технических наук *Ю. В. Тюменцев*

E-mail: tium@mai.ru

А. И. САМАРИН

НИИ нейрокибернетики им. А. Б. Когана
Ростовского государственного университета,
г. Ростов-на-Дону

E-mail: samarin@krinc.ru

НЕЙРОННЫЕ СЕТИ С ПРЕДНАСТРОЙКОЙ

Аннотация

Рассматривается задача динамического предпрограммирования нейронных сетей при реализации сложных функциональных зависимостей, заданных в неявном виде. Показывается, что введение свойств локальной адаптации позволяет существенно уменьшить число итерационных циклов в процессе поиска искомой зависимости, а распределенное представление функции позволяет заменить процедуру обучения процедурой преднастройки сети путем комбинаторного вытормаживания вставочных нейронов. Описывается использование такой сети с преднастройкой при решении задач идентификации объектов управления.

A. I. SAMARIN

A.B.Kogan Research Institute for Neurocybernetics,
Rostov State University,
Rostov-on-Don

E-mail: samarin@krinc.ru

NEURAL NETWORKS WITH PRELIMINARY ADJUSTMENT

Abstract

A dynamical preliminary adjustment problem is considered in regard to neural networks intended for realization of complex implicit functional dependencies. It is shown that involving of a local adaptation allow to reduce considerably a number of iterations during searching of the dependence. Moreover, distributed representation of the function enables to replace a training of the network with a preliminary adjustment by means of combinatorial inhibition of interneurons. An application of the network with preliminary adjustment is discussed concerning identification problem for control objects.

Введение

Современное состояние нейросетевых технологий может быть охарактеризовано как кризисное. Как и в сказке, перед витязем нейроинформатики в кризисной ситуации открываются три дороги.

Чтобы нейронные сети делали что-то полезное, решали широкий класс неформализованных задач можно выбрать три пути:

- нейронные сети должны обучаться решать эти задачи и главная проблема — это эффективные алгоритмы обучения;
- нейронные сети должны наращиваться под задачу и главная проблема — алгоритмы наращивания и технологии реализации растущих информационных структур;
- нейронные сети должны динамически поднастраиваться под решение текущей задачи и главная проблема — что и как поднастраивает, что и как поднастраивается.

Эти три направления развития нейросетевых технологий: *обучение, рост, подстройка* нейронных сетей имеют определенную и очень разную предысторию.

Наиболее богатую историю имеют нейронные сети с обучением, их становление проходило на примере решения прикладных задач, связанных с распознаванием образов. Это, прежде всего, работы *Ф. Розенблатта* 50-х годов 20-го века [1] и последующая лавина работ с 80-х годов по настоящее время, связанная с применением многослойных нейронных *сетей перцептронного типа* при решении проблемно-ориентированных задач распознавания образов, классификации и интерполяции функциональных зависимостей. Наглядным примером развития этого направления нейросетевых технологий на уровне отечественной науки могут служить проводимые ежегодно под руководством профессора *А. И. Галушкина* [2] научно-технические конференции «Нейрокомпьютеры и их применение». Рубрикация докладов этих конференций осуществлялась главным образом по области применения — от нейронных сетей в финансовых задачах до нейронных сетей в задачах химического производства.

К этому направлению — решение задач путем обучения, но уже в нейронных сетях с обратными связями, относятся нейросетевые технологии под общим названием *сети Хопфилда* [3]. Такие сети могут находиться в устойчивых и неустойчивых состояниях. Вектор входных воздействий выводит нейронную сеть из исходного состояния и она скатывается в другое

устойчивое состояние, характеризующееся локальным минимумом некоторой энергетической функции, определенной на значениях весов межнейронных связей. В зависимости от правила изменения этих весов — правила обучения, разные устойчивые состояния могут ассоциироваться с разными значениями вектора входных воздействий. Исследования этих сетей могут преследовать и другую цель, связанную с эффективной организацией динамической памяти.

Отдельной группой в задачах приложения нейронных сетей с обучением стоят исследования *семантических сетей ассоциативного типа*. В отечественных (советских) работах по нейроинформатике они представлены школой *Н. М. Амосова* [4] — это, прежде всего, работы *Л. М.* и *А. М. Касаткиных* [5] и *Т. Н. Байдык* [6].

Главные ограничения нейронных сетей с обучением:

- сеть, которая потенциально может все, в результате обучения реализует единственную функцию;
- обучение требует предварительного, достаточно трудоемкого формирования обучающей выборки;
- в силу глобальности алгоритмов обучения такие сети не могут дообучаться, а при увеличении обучающей выборки должны вновь обучаться на всей выборке.

Идея использования растущих или наращиваемых нейронных сетей связана в первую очередь с именем *С. Гроссберга*. Сети адаптивного резонанса (ART-сети) [7] главным образом преследовали цель снятия ограничений процедуры дообучения многослойных нейронных сетей. В такой сети, если по ходу функционирования появляются примеры, которые не укладываются в рамки функционирования уже обученной сети, то сеть наращивается новыми элементами, за счет которых и происходит дообучение без переобучения старой части сети.

Подобные же идеи растущих сетей рассматривались и академиком *А. Г. Ивахненко* [8] при анализе возможности сетевой реализации метода группового учета аргументов (МГУА) при адаптивной интерполяции функций.

Самая парадоксальная ситуация связана с проблемой динамической преднастройки нейронных сетей. Ее парадоксальность заключается в том, что в нейронных структурах мозга как нейрофизиологи, так и психологи считают наличие динамической преднастройки в реализации системных функций само собой разумеющимся. В моделях же, связанных с реализаци-

ей нейросетевых технологий, она практически не рассматривается. Идеология функциональной системы *П. К. Анохина* [9] предполагает наличие нейросетевого механизма акцептора результата действия, где ожидаемый результат сравнивается с реальным. При этом генерация модели ожидаемого результата осуществляется механизмом преднастройки. Практически все базовые положения когнитивной психологии [10] в процессе активного восприятия внешнего мира предполагают наличие такого механизма динамической преднастройки во время прогноза ожидаемого сенсорного сигнала. Одна из моделей реализации механизмов внимания [11] рассматривает внимание как некоторую процедуру адаптивной (динамически настраиваемой) фильтрации сенсорных сигналов. В электрофизиологии давно известно, что преднастройка электрографически сопровождается так называемой волной ожидания.

Практически все концепции, объясняющие реализацию мозгом системных функций, полагают, что одни нейронные структуры оказывают модулирующее или настраивающее влияние на другие структуры, но специалистами в области нейроинформатики эти механизмы практически не рассматриваются.

Следует отметить, что некоторые вычислительные технологии начала семидесятых годов базировались на идеях использования перепрограммируемых или перенастраиваемых вычислительных структур.

Наиболее известные модели описаны в монографии *Э. В. Евреинова* и *И. В. Прангишвили* «Цифровые автоматы с настраиваемой структурой» [12] и представляли собой однородную среду вычислительных или логических элементов, коммутируемых под решение конкретной задачи. Авторы отмечали, что благодаря однотипности элементов и связей между ними существенно повышается технологичность при применении методов интегральной технологии, повышается надежность за счет возможности перекоммутации среды при выходе из строя части элементов, повышается скорость решения задач за счет параллельности вычислений. Идеи эти не получили достаточного развития, видимо, в связи с тем, что каждый из элементов однородной среды реализовал простейшую функцию и всю среду нужно было очень сложно программировать. В итоге все выгоды перекрывались сложностями программирования структур.

Значительные успехи были достигнуты школой *А. В. Каляева* [13] при создании многопроцессорных коммутируемых вычислительных структур. Процессоры в такой системе отличаются от стандартных, так называемых фон Неймановских, тем, что они должны иметь очень широкую систему

машинных команд и одноканальную реализацию процессорных функций, т. е. после подачи входных величин независимо от того, какая функция реализуется в данный момент — сложение, произведение или извлечение корня, через одинаковое время Δt должно быть получено выходное значение функции. Множество таких макропроцессоров образуют динамические вычислительные структуры путем использования специально разработанных коммутирующих регистровых структур. *А. В. Каляев* отмечал, что наиболее перспективной для реализации таких макропроцессоров может быть нейросетевая технология.

В плане ретроспективы развития нейрокибернетических идей можно сказать, что исходно к развитию работ, связанных с динамической настройкой нейронных сетей, ближе всего были исследования школы *У. Мак-Каллока*. Но исследования эти почему-то довольно рано оказались вытеснены прагматически более эффективными работами, связанными с процедурами обучения сети на реализацию единственной функции.

Ссылки на *У. Мак-Каллока* в нейросетевой литературе встречаются, пожалуй, столь же часто, как и ссылки на *Д. Хебба*, но в обоих случаях это ссылки на единственные работы [14, 15], а в отношении *У. Мак-Каллока* эти ссылки еще и некорректны. Поэтому прежде чем описывать класс задач, в которых без введения свойств преднастройки нейронной сети невозможно добиться эффективного решения, коротко остановимся на наследии *У. Мак-Каллока*.

Нейронные сети У. Мак-Каллока

Как упоминалось, большинство работ, в которых рассматриваются сети из формальных нейронов, ссылаются на описание свойств нейрона, данное в работе *У. Мак-Каллока* и *У. Питтса* 1943 года [14]. Главная проблема, которой посвящена эта работа, заключается в рассмотрении и доказательстве возможности реализации произвольной логической функции алгебры высказываний путем построения структуры из унифицированных элементов. Свойства этих элементов напоминают известные свойства нервных клеток. В модели такого нейрона осуществляется арифметическое суммирование возбуждающих воздействий и при превышении суммы порогового значения нейрон переходит в возбужденное состояние, если же на нейрон приходит тормозное возмущение, то оно действует абсолютно и переводит нейрон в состояние с нулевым выходом. Из таких нейронов и нужно было синтезировать (растить) сеть, которая реализовала бы любую искомую функцию.

В последующих работах, ключевой из которых является работа с подзаголовком Agathe Tuche (счастливая находка) [14], Мак-Каллок вводит другие свойства нейрона и исследует уже проблемы, в прямую не связанные с вопросами функциональной полноты нейронного базиса.

В работе [15] постулируются свойства формального нейрона Мак-Каллока. *Формальный нейрон* — элемент вычислительной машины, обладающий следующими свойствами:

1. К нему подходят нервные волокна от δ входов, а сам он имеет один выход.
2. Каждый вход и единственный выход могут находиться только в двух состояниях: возбужденном или невозбужденном.
3. Нервные волокна от некоторого входа могут разветвляться, но не могут объединяться с волокнами другого входа.
4. Волокна могут быть возбуждающими и тормозными. По возбуждающему волокну нейрон получает положительную единицу возбуждения (+1), а по тормозящему — отрицательную (–1). Волокно может также запереть сигнал, идущий по другому волокну (рис. 1).
5. Сигналы могут проходить через нейрон только в одном направлении.
6. Имеется некоторое запаздывание в передаче сигнала от входа к нейрону.
7. При правильной работе нейрон возбуждается, если алгебраическая сумма возбуждающих и тормозящих сигналов превосходит некоторый определенный порог (Θ).

Такое подробное описание свойств формального нейрона Мак-Каллока приводится для того, чтобы обратить внимание на некоторую некорректность, допускаемую в очень многих работах. Эта некорректность заключается в следующем:

- при постулировании алгебраического суммирования входных воздействий некорректно ссылаться на работу 1943 года Мак-Каллока и Питтса, так как в этой работе постулируется абсолютное торможение;
- часто встречающаяся ссылка на то, что однослойная сеть формальных нейронов Мак-Каллока реализует только линейно разделимые функции (проблема исключенного ИЛИ), также некорректна, так как при наличии запрещающих волокон единственный формальный нейрон Мак-Каллока может реализовать произвольную логическую функцию от δ переменных [16].

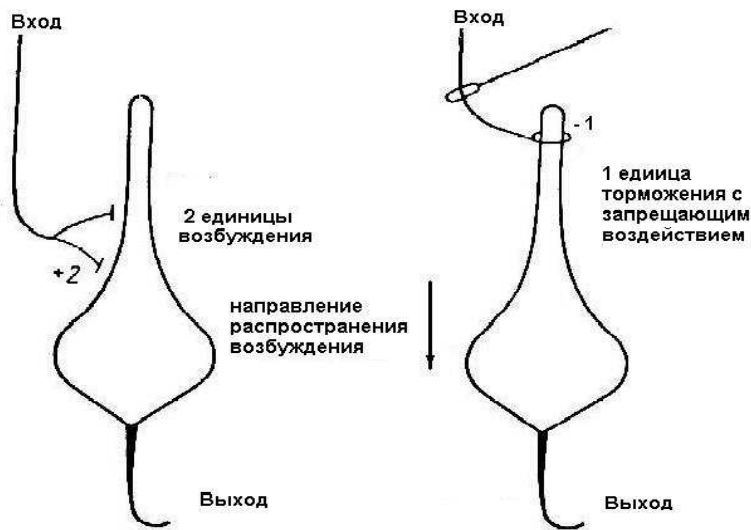


Рис. 1. Схематическое изображение формального нейрона и типы входных волокон [15]

У. Мак-Каллок рассматривал формальный нейрон прежде всего как элемент вычислительной машины. Основные его работы были посвящены проблеме надежности машины, построенной из таких элементов. Мак-Каллок и его коллеги рассматривали изменения функций нейросетевых конструкций при независимом и сопряженном изменении порогов нейронов сети при изменении сигналов, связанных с изменением синаптических коэффициентов. В общем, в исследованиях 50-х годов его интересовала не функциональная пластичность нейронной сети, а способы борьбы с этой пластичностью [17].

Для описания логических функций, реализуемых нейроном и сетью, им было введено в практику символическое изображение функций, имеющее наглядную геометрическую интерпретацию для нейронов и сетей с ограниченным числом входов ($\delta \leq 5$) (рис. 2).

Фактически диаграмма Венна — это модифицированная таблица истинности, с помощью которой записывается логическая функция, заданная в совершенной дизъюнктивной нормальной форме (СДНФ). Если на комбинацию входных воздействий, соответствующей какой-то зоне диаграммы,

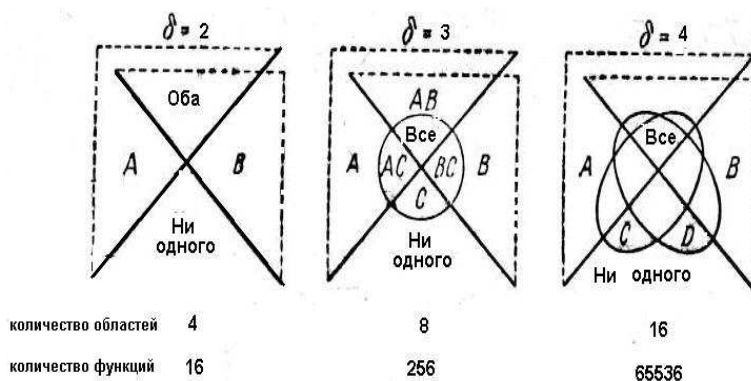


Рис. 2. Символы Венна для записи логических функций, реализуемых формальным нейроном (δ — количество входных переменных) [18]

нейрон возбуждается, то в этой зоне диаграммы ставится точка. При определенном пороге нейрон может быть возбужден при нескольких комбинациях входных воздействий, тогда на диаграмме будет проставлено несколько точек. Общее количество возможных функций δ переменных равно 2^{2^δ} .

Нейрон, который реализует максимальное число функций при изменении его порога от максимального до минимального значения, называется невырожденным нейроном. У такого нейрона при последовательном уменьшении порога на единицу в диаграмме Венна появляется новая точка. Пример двух таких нейронов для $\delta = 3$ приведен на рис. 3. Первый нейрон не имеет запрещающих волокон и при изменении порога на единицу последовательно реализует $2^\delta + 1$ линейно разделимых функций. Второй нейрон содержит запрещающие волокна, поэтому позволяет реализовать и нелинейные функции. Так, при пороге $\Theta = +1$ он реализует очень важную для функционально универсальных сетей функцию \oplus — сложение по модулю два.

Из таких невырожденных нейронов синтезировалась сеть Мак-Каллока, в которой все нейроны первого ряда получали воздействия от всех входов, нейроны второго ряда — от всех выходов первого ряда. Последний, выходной, слой сети содержал один элемент. В сети отсутствуют обратные связи. У Мак-Каллок исследовал возможность повышения надежности функционирования такой сети при использовании так называемой избыточности

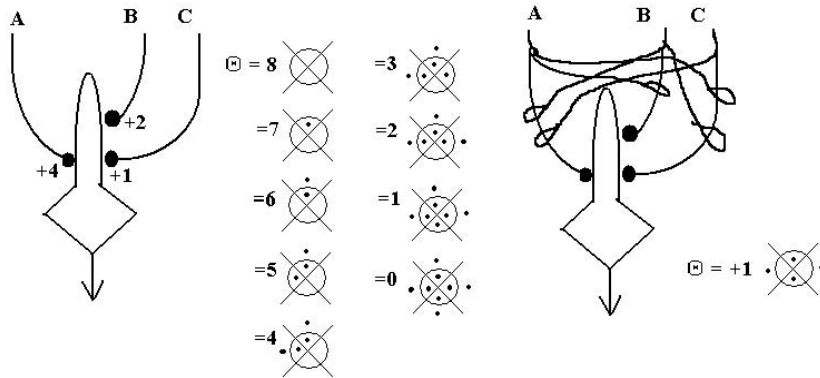


Рис. 3. Невырожденные формальные нейроны

вычислений. В нейросетевом приложении такая избыточность заключается в том, что изменение функций одних нейронов при изменении их порогов может быть скомпенсировано изменением функции других нейронов, в результате чего общая функция сети не изменится.

Учениками У.Мак-Каллока [19] было доказана важная теорема, согласно которой избыточность вычислений не сводится к избыточности кода или канала и что только избыточность вычислений приводит к достижению требуемой надежности без потери пропускной способности системы.

Таким образом, для актуальных задач того времени (надежность информационно-вычислительных систем) школой Мак-Каллока было получено казалось бы парадоксальное заключение, что стабильность вычислений в нейронных сетях может быть достигнута за счет пластичности ее элементов. В двух работах [18, 20] У.Мак-Каллок отмечал, что функциональная пластичность или функциональная универсальность может давать и другое преимущество нейросетевых технологий. Было показано [21], что при очень малой избыточности элементов только за счет изменения порогов, не переконструируя сеть и не меняя весов связей, можно настроить сеть на реализацию любой логической функции от заданного числа переменных, то есть сеть становится функционально полностью универсальной и динамически программируемой. Если же в сеть вводить обратные и перекрестные связи, то с учетом постулируемых временных задержек можно синтезировать сети каскадного типа, которые реализуют временные булевы

функции [22].

Таким образом, исследования школы У. Мак-Каллока, в общем-то косвенно, привели к постановке задачи синтеза динамически программируемых нейронных сетей. Варьируемым параметром при программировании или преднастройке таких сетей может выступать порог нейрона, являющийся аддитивным параметром в его описании. В качестве такого управляемого порога могут выступать воздействия, поступающие на данную нейронную сеть от другой сети, связанной с другими переменными. В этом случае можно говорить, что одна нейронная сеть преднастраивает другую нейронную сеть на ту или иную интерпретацию ее специфических входных сигналов.

Как интерпретировать входные сигналы в той или иной ситуации?

Почему при использовании нейронных сетей в качестве информационно-управляющих систем практически всегда возникает неоднозначность интерпретации входных воздействий в условиях замыкания обратной связи через окружение?

Общая постановка задачи

Формальная постановка задачи поведения во внешней среде системы с целеполаганием представляется следующим образом. Априорно неизвестная среда задается совокупностью m параметров:

$$E(t) = (e_1, e_2, \dots, e_m),$$

управляемая система задается совокупностью n параметров:

$$S(t) = (s_1, s_2, \dots, s_n).$$

Система оснащена сенсорными органами разной модальности и в результате взаимодействия системы S со средой E она получает разные сенсорные воздействия: визуальные $V = V(S, E)$, осязательные или силовые $F = F(S, E)$, возможно, слуховые, температурные и др. Функционирование или поведение системы определяется управляющим воздействием $U(t)$, т. е. $S(t) = S[U(t)]$. Целью управления является получение оптимального поведения системы, при котором траектория $S(t)$ доставляет оптимум некоторому критерию $Q = Q[V(t), F(t)]$. На понятийном уровне такая цель дается в виде предписания, например: двигаться так, чтобы в поле зрения попал такой-то объект, или двигаться к целевому объекту и не столкнуться

с другими объектами внешней среды. Таким образом, нужно найти управление переменными, связанными с перемещением системы в среде, как функцию от сенсорных переменных $U(t) = U(V, F)$, при котором

$$Q = Q[V(S(U), E(t)), F(S(U), E(t))] \rightarrow Q_{opt}.$$

Особенностью описанной модели взаимодействия системы со средой является то, что сенсорная информация и критерий управления существенным образом зависят как от состояния среды, так и от состояния системы. Следовательно, текущая идентификация среды, а значит и эффективное управление поведением системы в этой среде, невозможна без идентификации самой управляющей системы. Возникающая проблема идентификации собственно системы обусловлена следующими обстоятельствами. В силу относительности движения любой изменяющийся сенсорный сигнал невозможно однозначно проинтерпретировать. За счет чего вызвано это изменение — является ли оно результатом движения системы в неизменной среде, или же результатом влияния динамически меняющейся среды на неподвижную систему, или же результатом действия обоих факторов. Для однозначной интерпретации сенсорных воздействий, а, следовательно, и управляемости системы, должен существовать механизм, который позволяет определить какая часть изменения сенсорного сигнала связана с собственным движением системы, тогда оставшаяся часть изменения связана с динамическим изменением среды. Такой механизм обеспечивает свойство константности восприятия, присущее живым системам, и является пока почти неразрешимой проблемой при создании автономных систем с активным поведением во внешней среде.

Поскольку состояние среды E заранее не известно, то искомое управление U не может быть найдено аналитически, более того, нельзя даже заранее определить вид требуемой функции. Закон управления может быть существенно нелинейным, включать в себя аналоговые и дискретные компоненты, а также иметь прямую зависимость от времени. Поэтому при выборе класса функциональных преобразований, в рамках которого ищется закон управления, трудно вводить какие-либо ограничения. Из работ по нейроинформатике известно [24–26], что сети динамических нейронов образуют максимально полный базис функциональных преобразований. Тогда для поиска закона управления необходимо синтезировать сеть, свойства элементов и топология связей которой заранее обеспечивают способность к очень широкому классу функциональных преобразований, а затем, путем

адаптации, подстройки или обучения сети, найти требуемое преобразование, отвечающее системному критерию поведения системы.

В силу изменчивости окружающей среды, характерным для решения поведенческих задач становятся непрерывные цепи изменяющихся целей поведения, что требует от системы управления обеспечения динамической смены функций, необходимых для осуществления процессов управления на разных этапах поведения системы, с использованием одной и той же структуры.

В условиях не полностью известной среды модели сенсоров, системы в целом и ее окружения будут содержать множество неизвестных параметров. В этом случае реализация управления на основе программных алгоритмов не представляется возможной. Поэтому при создании объектов с целенаправленным поведением – роботов, существенное значение приобретают структуры и алгоритмы, которые обеспечивают адаптивные свойства системы управления. Вся необходимая информация, которая недоступна в форме априорных знаний, должна быть приобретена управляющей системой робота в процессе активного взаимодействия со средой, т. е. такая система с необходимостью должна быть обучающейся.

Важное место среди обучающихся структур занимают искусственные нейронные сети [24]. Однако накопленный к настоящему времени опыт решения задач, связанных с организацией поведения роботов, использующих информацию об окружении, показывает, что проблема координированного управления многоразовными роботами не имеет удовлетворительного решения в рамках существующих нейросетевых парадигм. Такая ситуация заставляет исследователей искать другие пути, но предлагаемые при этом структуры не позволяют решать сколько-нибудь приближенные к реальности задачи. Это относится, например, к создаваемым системам управления для наиболее продвинутых гуманоидных роботов. Используемые в них различные нечеткие (fuzzy) алгоритмы основаны на простых, интуитивно понятных правилах обучения, но сценарии поведения жестко ограничиваются этим набором правил [27].

Существуют устоявшиеся рецепты решения задач управления поведением в условиях неполной или противоречивой информации:

- неполная информация об объекте управления и среде – применение алгоритмов адаптации [28],
- противоречивая или неоднозначно интерпретируемая информация – применение алгоритмов нечеткой логики [29].

В то же время многообразие алгоритмов адаптации достаточно велико и не всегда эффективно, а условия противоречивости и неопределенности имеют в приложении к роботам и другую альтернативу — активное действие в среде, направленное на снятие этого противоречия или неопределенности [25].

При управлении поведением робота адаптация, как правило, достигается путем подстройки параметров, заранее выбранных алгоритмов. В силу функциональных ограничений этих алгоритмов невозможно выйти за их пределы и найти принципиально иное решение задачи, а от интеллектуальной системы ожидаются именно такие решения.

Адекватность задачам сенсомоторного поведения именно нейросетевых реализаций не вызывает сомнения, так как мозг человека и животных успешно их решает, но эффективного решения таких задач пока не было предложено. Поэтому построение адаптивных структур на основе нейроподобных элементов остается наиболее актуальным аспектом при реализации информационно-управляющей системы робота.

Неоднозначность интерпретации сенсорной информации в условиях замыкания обратной связи через окружение

Так как в процессе поведения робота происходит замыкание обратной связи через окружение, то само перемещение робота в пространстве приводит к изменению сенсорных воздействий. Одновременно к изменению сенсорных воздействий могут приводить и воздействия внешней среды. В таких условиях вероятно возникновение ситуаций, когда сенсорная информация не может интерпретироваться системой однозначно, а значит и управление роботом не может быть однозначно организовано.

Для снятия этой неоднозначности в интерпретации сенсорных сигналов используется поисковая стратегия как основной способ организации управления подобными системами с обратной связью. В процессе активного взаимодействия робота с внешней средой система управления обучается распознавать, какая часть сенсорного сигнала связана, например, с изменением позы манипулятора в поле тяготения, а какая с действием внешних факторов (рис. 4). На рис. 4-а и 4-с объект в схвате находится под действием только силы тяжести, определенной конфигурации манипулятора (θ_1, θ_2). Каждой позе соответствует определенное разложение вектора силы тяже-

сти схвата с нагрузкой на компоненты (F_1, F_2) датчика силы. На рис. 4-b, объект в схвате находится под действием силы тяжести и внешней силы, при этом сенсорный сигнал такой, что соответствует конфигурации случая 4-с, а реальный вектор состояния соответствует 4-а. В данной ситуации без правильной идентификации управляемой системы подходящее управляющее воздействие $U(\Theta, F)$, обеспечивающее решение поставленной задачи, не может быть найдено.

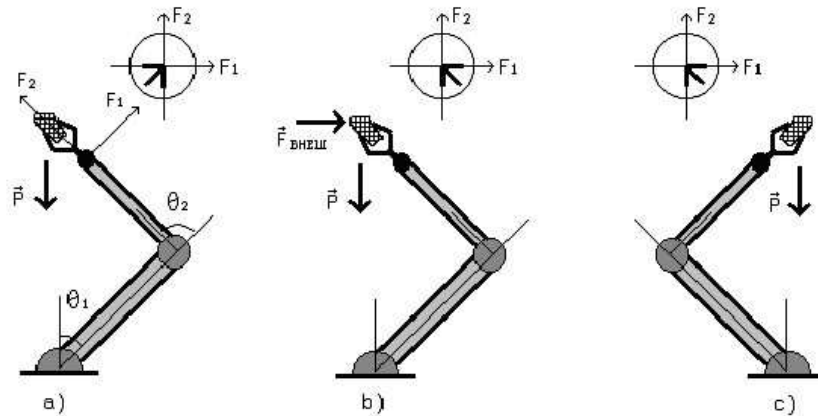


Рис. 4. Неопределенность ситуации в случае действия внешней силы

Следовательно, текущая идентификация среды, а значит, и эффективное управление поведением системы в этой среде, невозможны без идентификации самой управляемой системы.

Нейронные сети с преднастройкой функциональных преобразований

Предлагаемый подход является, в некотором смысле, развитием идей У. МакКаллока о полифункциональных нейронных сетях логических функций, когда путем изменения порогов элементов можно динамически преднастроить сеть на реализацию любой из них. В данной работе также рассматриваются полифункциональные нейронные сети, в которых те или иные функциональные преобразования реализуются преднастройкой некоторых управляемых параметров.

Если синтезировать нейронную сеть с очень широким диапазоном функциональных преобразований, то решение поставленной задачи будет заключаться в поиске преднастраивающего воздействия на управляемые параметры сети для получения конкретной требуемой функции. В этом случае в процессе адаптации, реализуемой посредством процедуры преднастройки, не происходит потеря функциональной универсальности в отличие от традиционных нейронных сетей, в которых обучение приводит к реализации единственной функции.

Процедура преднастройки, как и процедура обучения, нейронной сети на реализацию требуемой функции может достигаться двумя путями. Первый путь связан с подстройкой или обучением параметров, влияющих на вид функциональной зависимости в целом. Такие алгоритмы используются в многослойных сетях перцептронного типа и определяют единые (не меняющиеся) значения параметров на всей области определения функции. Главным недостатком, ограничивающим применение этих сетей, как отмечалось во введении, является именно глобальность адаптации, которая при обучении требует многократного возвращения к одним и тем же примерам обучающей выборки, а при дообучении в локальной области требует фактически нового обучения с доступом ко всем ранее использованным примерам. Как также отмечалось во введении, в качестве пути устранения этого недостатка *С. Гроссбергом* и *Д. Карпенгер* [7] были предложены сети адаптивного резонанса (ART-сети). Особенности алгоритмов, реализуемых такими сетями, заключается в использовании процедур локальной адаптации. Суть такой локальной адаптации заключается в наращивании нейронной сети для каждого нового примера, не относящегося к ранее тестированным областям определения искомой функции. Такие сети растут, при появлении новых примеров не перестраиваются, а достраиваются. В итоге же, как и для сетей перцептронного типа, в результате обучения сеть реализует единственную функцию и не обладает способностью динамически преднастраиваться на реализацию той или иной функции, но все же при обучении новому, старое не забывает, и при этом не требует обязательного сохранения всей обучающей выборки.

По *У. Мак-Каллоку* [18] именно функциональная универсальность при относительно небольшой избыточности сети наиболее привлекательна в биологически правдоподобных нейронных сетях. Только такие функционально универсальные, динамически перенастраиваемые, «программируемые» на реализацию текущей функции нейронные сети на самом деле могут считаться нейрокомпьютерами.

Разработанные в НИИ нейрокибернетики им. А. Б. Когана для робототехнических приложений нейронные сети с преднастройкой свободны от недостатков изложенных нейросетевых парадигм [30]. Они могут дообучаться по ходу функционирования в составе реальной системы управления, используют алгоритмы локальной адаптации и при некоторой распределенной избыточности элементов позволяют менять параметры аппроксимируемой функции не за счет обучения или наращивания этой сети, а за счет комбинаторного вытормаживания «вставочных» нейронов другой сетью. Идеи, использованные при разработке такой нейронной сети, базировались на нейрофизиологически непротиворечивых соображениях и имели достаточно выраженный инженерный характер с позиций возможной аппаратной реализуемости без использования компьютерных эмуляций.

В качестве базового элемента нейронной сети выбран нейрон со следующей активационной функцией:

$$u = x + \sum_j w_j y_j - h,$$

$$y = Q(u),$$

где u — мембранный потенциал нейрона; x — внешнее входное воздействие на нейрон; w_j — весовой коэффициент связи от нейрона j ; y_j — выходная функция нейрона j , имеющая смысл частоты импульсной активности; h — порог нейрона; k — коэффициент усиления; $Q(u)$ — нелинейная активационная функция мембранного потенциала:

$$Q(u) = \begin{cases} 1, & \text{если } ku \geq 1; \\ ku, & \text{если } 0 < ku < 1; \\ 0, & \text{если } ku \leq 0. \end{cases}$$

Для реализации кусочно-непрерывной аппроксимации предложена ячейка нейронной сети, составленная из скрытого нейрона 3, вставочных нейронов 1 и 2 (рис. 5).

Различные веса тормозных связей со вставочных нейронов позволяют получать произвольные выходные функции из выбранного класса. В данном случае этот класс ограничен линейными функциями, что было определено при выборе кусочно-линейной активационной функции нейронов сети.

Если для каждого интервала области определения кусочно-линейной

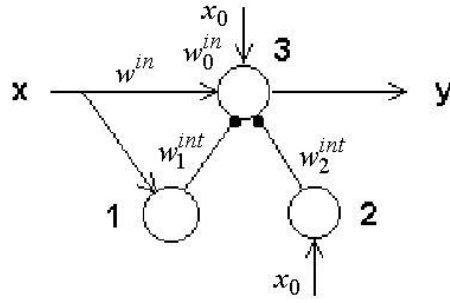


Рис. 5. Элементарная функциональная ячейка нейронной сети

аппроксимации функции

$$y = (w^{in} - w_1^{int})x + (w_0^{in} - w_2^{int})x_0;$$

$$K = w^{in} - w_1^{int}; \quad c = w_0^{in} - w_2^{int};$$

$$\hat{y} = Kx + c$$

выбирать свое значение K и c и переключать эти значения при переходе в разные участки области определения, то таким образом может быть реализована любая функция. Значения этих коэффициентов определяются в свою очередь значениями весовых коэффициентов нейронов элементарной базовой ячейки и весь вопрос заключается в том, каким образом эти значения находить для каждого интервала области определения.

Предлагается, за счет введения некоторой избыточности вставочных нейронов в базовую ячейку, получать требуемые значения коэффициентов путем комбинаторного их вытормаживания.

Система уравнений, описывающих нейронную сеть кусочно-линейной аппроксимации, может быть представлена в следующем виде:

$$u^{hid} = \sum_{j=0}^{n_{int}^1} w_j^{in} x_j - \sum_{j=1}^{n_{in}} w_j^{int} y_j^{int},$$

$$u_i^{int} = \sum_{j=0}^{n_{in}} w_{ij}^{in} x_j - p_i, \quad i = 1, \dots, n_{int},$$

$$y = Q(u),$$

где u^{hid} — мембранный потенциал нейрона скрытого слоя; u^{int} и y^{int} — мембранный потенциал и активность вставочных нейронов; $n_{int} = n_{int}^1 + n_{int}^2$ — общее количество вставочных нейронов; $p = P(x) = (p_1, p_2, \dots, p_n)$ — вектор преднастройки.

Для выделенной группы нейронов определяются возбуждающие связи с соответствующего входа сети для получения необходимого коэффициента крутизны. Следует заметить, что аппроксимация величины смещения, реализуемой в нейросети линейной функцией, осуществляется весами тормозных связей фоновоактивных вставочных нейронов.

Активационная функция нейронов сети с кусочно-линейным поведением позволяет переводить вставочные нейроны в состояние с нулевой активностью посредством сильного внешнего вытормаживающего сигнала — «выключать» их. Введение комбинаторики «включенных» и «выключенных» вставочных нейронов при увеличении количества вставочных нейронов с некоторым образом определенными величинами тормозных воздействий приводит к распределенному представлению выходных функций в структуре нейронной сети.

Множество комбинаций $P(x)$ вытормаживания вставочных нейронов позволяет реализовать множество различных кусочно-линейных функций. При этом разные комбинации «выключения» вставочных нейронов на различных участках области определения приводят к тому, что суммирование по коэффициентам весов тормозных связей активных вставочных нейронов, выполняемое скрытым нейроном сети, будет давать параметры прямой, являющейся аппроксимацией исходной функции для некоторого участка (рис. 6).

Как видно из рис. 6, вставочные нейроны условно группируются для аппроксимации коэффициентов крутизны (1) и величины смещения (2) аппроксимирующей линейной функцией. Для данной сети, которую мы будем называть *нейронной сетью адаптивной фильтрации*, предполагается, что веса тормозных связей выбраны так, что всегда можно найти комбинацию активных вставочных нейронов, суммирование по весам тормозных связей которых дает необходимые параметры для получения аппроксимирующей кусочно-линейной функции, удовлетворяющей заданному условию точности аппроксимации на текущем участке исходной функциональной зависимости.

Для каждого следующего участка аппроксимации ищется своя комбинация вставочных нейронов; при этом нейроны, не задействованные в данной комбинации, полностью вытормаживаются. Процедура вытормажива-

ния вставочных нейронов сети адаптивной фильтрации называется преднастройкой. Следовательно, можно говорить о преднастройке нейронной сети на каждом локальном участке на требуемое функциональное преобразование.

Нейронная сеть способна работать с функциями нескольких переменных посредством увеличения количества вставочных нейронов (рис. 7).

Основная идея состоит том, что построение требуемого отображения происходит без изменения топологии и весов связей сети, выбранных в процессе ее реализации. Поэтому структура и параметры нейронной сети выбираются так, чтобы гарантировать необходимую широту производимых этой сетью функциональных преобразований.

Таким образом, предполагается существование второй нейронной сети — *сети преднастройки (состояний)*, которая реализует поиск вытормаживающей комбинации p на вставочные нейроны сети адаптивной фильтрации и ее запоминание для данного участка аппроксимации. Фактически, нейронная сеть преднастройки реализует функцию $p = P(x)$, где вход x может в общем случае и не совпадать с входным сигналом нейросети адаптивной фильтрации, что является важным для решения задач сенсомоторной координации, требующих взаимодействия сенсорных потоков разной модальности.

В данной работе конкретная реализация нейронной сети преднастройки не рассматривается. Каждый выход этой сети заканчивается на отдельном вставочном нейроне первой сети и при своей активности полностью тормозит связанный с ним вставочный нейрон. Комбинаторное вытормаживающее воздействие на вставочные нейроны может быть представлено вектором-столбцом $P(x) = (p_1, p_2, \dots, p_n)^T$. Тогда матрица весовых коэффициентов нейронной сети адаптивной фильтрации $W' = W'(P, W)$ определяется текущим выходом сети преднастройки $p = P(x)$ и весовой матрицей W , составленной из весов тормозных связей между скрытым слоем и слоем вставочных нейронов сети. Таким образом, преднастраивающая функция сети состояний осуществляется реализацией набора матриц весовых коэффициентов W' .

Повышение размерности задачи требует введения соответствующего количества дополнительных эквивалентных нейронных сетей, работающих в параллельном режиме, что согласуется с известными принципами параллельности вычислений в нейронных сетях.

Инициализация весовых коэффициентов случайными равномерно распределенными значениями является наиболее простой и не требующей до-

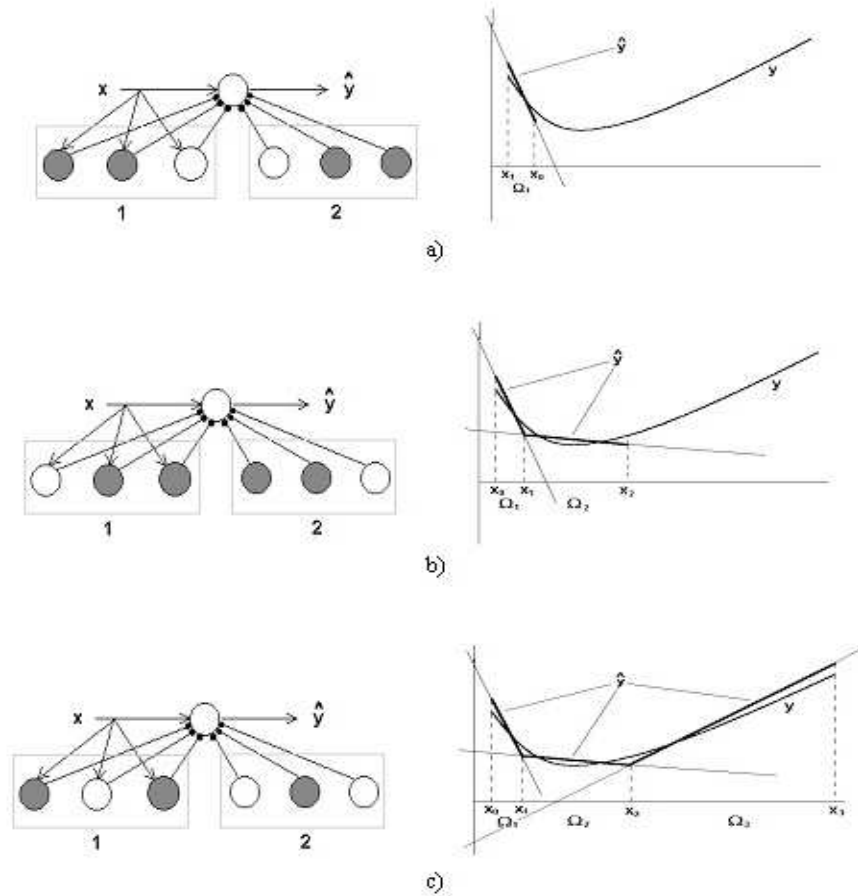


Рис. 6. Кусочно-линейная аппроксимация в нейронной сети реализуется нахождением комбинаций включенных и выключенных вставочных нейронов для разных подобластей. Серым цветом обозначены выключенные нейроны.

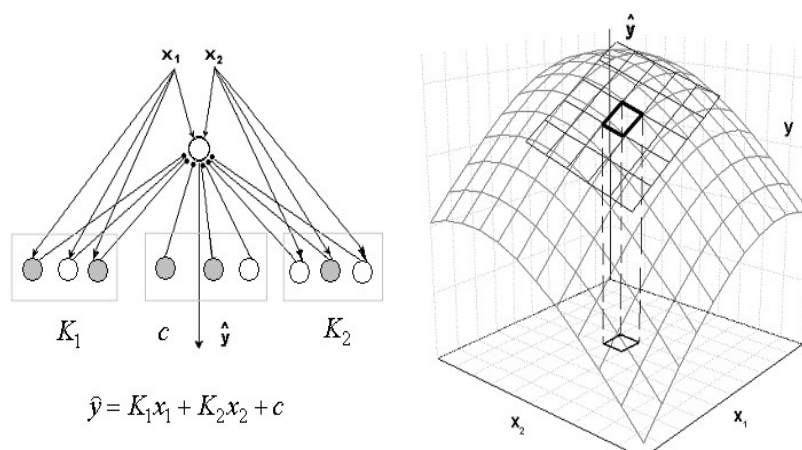


Рис. 7. Архитектура нейронной сети при аппроксимации функций двух переменных

полнительных вычислительных процедур. Кроме того, получаемая небольшая избыточность в реализуемых функциях обеспечивает надежность при поиске необходимой вытормаживающей комбинации.

Вычислительные эксперименты показывают, что нейронная сеть небольших размеров демонстрирует достаточно высокую точность аппроксимации. Так, например, 256 градаций значений коэффициента аппроксимирующей функции при инициализации случайными значениями получается при 12–13 вставочных нейронах, тогда как для расчетных значений — это 8 вставочных нейронов, а для одинаковых значений — 256.

Простота и эффективность реализации нейронной сети со случайными весами связей позволила выбрать этот вариант в качестве рабочего и использовать его далее в проводимых исследованиях.

Процедура преднастройки в разработанной сети заключается в поиске вытормаживающих комбинаций вставочных нейронов для построения необходимой функциональной зависимости. Общее предположение состояло в том, что никакая априорная информация для реализации алгоритмизированного поиска недоступна. В таких случаях обращаются к методам перебора, как наиболее универсальным методам поиска решений. Среди рассмотренных эволюционного, последовательного и случайного методов

выбран последний метод перебора, как наиболее эффективный при поиске комбинаций вытормаживания вставочных нейронов.

Исследования зависимости точности аппроксимации в нейронной сети с преднастройкой со случайным поиском вытормаживающих комбинаций от количества вставочных нейронов проведены на тестовых функциях $y(x)$, представляющих собой синусоиды с различными фазовыми сдвигами.

Построение аппроксимации $\hat{y}(x)$ данных функций позволяет протестировать нейронную сеть в широком диапазоне реализуемых коэффициентов крутизны и смещений аппроксимируемых функций. В результате тестирования получена зависимость точности аппроксимации от количества итераций для нахождения требуемого вектора преднастройки при разном числе вставочных нейронов (рис. 8).

В экспериментах показано, что время поиска нужной комбинации намного меньше общего количества возможных их вариантов и, кроме того, с увеличением количества вставочных нейронов число итераций преднастройки для поиска очередной вытормаживающей комбинации имеет тенденцию уменьшаться за счет большей избыточности в реализуемых функциях.

Выбранный алгоритм преднастройки выступает в качестве адаптивной процедуры построения функциональной зависимости в сети. Комбинация вытормаживания p вставочных нейронов для получения аппроксимирующей функции \hat{y}_p некоторого участка исходной зависимости y ищется случайным образом. Как только находится комбинация, которая удовлетворяет условию точности аппроксимации ε , она запоминается. Для каждого следующего участка аппроксимации Ω_p ищется своя вытормаживающая комбинация \mathbf{p}_p :

$$\mathbf{p}_p \in \Omega_p : |y - \hat{y}_p| < \varepsilon, \Omega = \bigcup_p \Omega_p.$$

Однако пока текущая аппроксимирующая функция удовлетворяет условию точности аппроксимации, смена вектора преднастройки не происходит.

В этом заключается один из ключевых моментов аппроксимации в модели. Так как дискретизацию области определения нельзя провести заранее из-за недоступности данных в рассматриваемых задачах, то нейронная сеть преднастройки реализует таковую непосредственно в процессе построения аппроксимации и каждый участок кодирует распределением активности своих выходных нейронов.

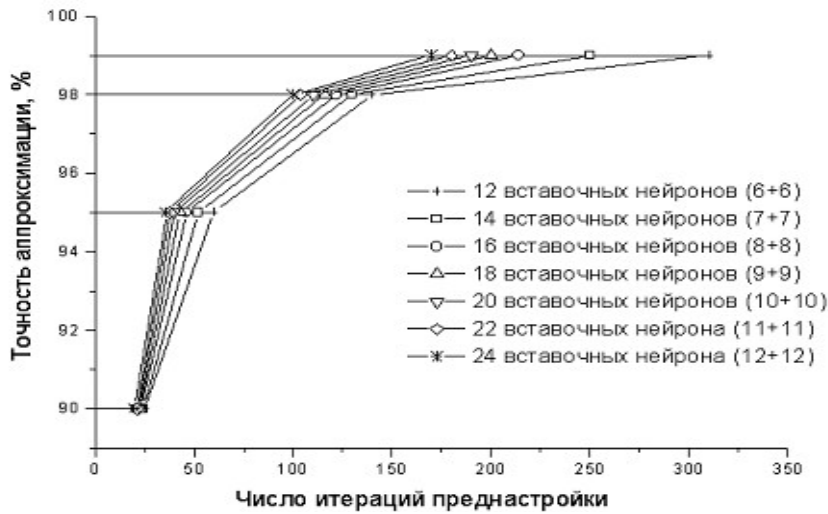


Рис. 8. Зависимость точности аппроксимации от числа итераций (поискowych шагов) для нахождения очередного вектора преднастройки. Запись $(n + n)$ означает, что для приближения каждого параметра аппроксимирующей функции выбраны по n вставочных нейронов.

Результаты моделирования подтвердили предположение о том, что разработанная модель нейронной сети является состоятельной для решения задач отображения произвольных функциональных зависимостей.

Следует отметить, что для сети с преднастройкой характерны, как и для всех сетей с локальной аппроксимацией, слабые обобщающие свойства, если обучающих примеров недостаточно много. Однако эффективность решения рассматриваемых в данной работе задач не сводится только к формированию обучающей выборки. Оценка правильности функционирования нейронной сети производится некоторым критерием в каждый момент времени. Например, если данный критерий задан на сенсорных переменных, то обратная связь обеспечивает поток этих данных непрерывно. Следовательно, в общем случае можно говорить, что для нейросети с преднастройкой, реализованной для решения такого рода задач, проблема обобщающих свойств не является принципиальной.

Среди ключевых моментов, выявленных при исследовании нейронных

сетей с преднастройкой, можно выделить следующие:

- локальные аппроксимирующие свойства нейронной сети позволяют осуществлять процедуры перенастройки на частично измененную исходную функциональную зависимость только на требуемых участках;
- распределенный характер представления реализуемых функций способствует достижению высокой точности аппроксимации сложных зависимостей при небольших размерах нейронной сети;
- другой, чем в существующих нейросетевых парадигмах, механизм построения необходимого функционального преобразования позволяет добиться независимости структуры нейронной сети и ее параметров от вида конкретной аппроксимируемой функции.

Нейросетевые механизмы константности восприятия при формировании сенсомоторной координации робота

Описываемая модель нейронной сети с преднастройкой в прикладном плане тестировалась на примере реализации функции константности восприятия в системе управления робота-манипулятора. Суть проблемы и ее решение демонстрируются на программной модели двухзвенного плоского манипулятора, оснащенного двухкомпонентным датчиком силы в запястье схвата (рис. 9).

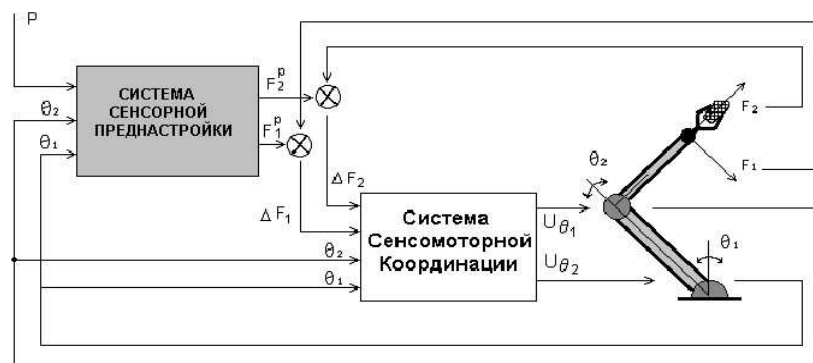


Рис. 9. Структурная схема системы «манипулятор-среда-управление»

Предполагается, что существуют или уже найдены алгоритмы управления манипулятором, реализующими функции реагирования манипулятора на внешнее воздействие (на блок-схеме рис. 9 эти алгоритмы реализуются системой сенсомоторной координации). В зависимости от целей управления это может быть управление, обеспечивающее уход манипулятора от внешних действующих сил (так называемая искусственная податливость), или же управление, обеспечивающее воздействие внешней силы строго по нормали к поверхности, с которой взаимодействует объект манипулирования (задача вставления стержня в отверстие с малым зазором).

Кроме силы, вызванной взаимодействием объекта манипулирования с объектами внешней среды, силоизмерительными сенсорами регистрируются еще и гравитационные составляющие, зависящие от массы объекта в схвате и позы манипулятора. Причем по абсолютной величине эти воздействия могут на порядок превосходить значения внешних сил, относительно которых строится закон управления. Для выделения этих малых сигналов на высоком уровне фоновых воздействий используется дополнительный блок системы управления (на рис. 9 выделен серым цветом) — система сенсорной преднастройки. Именно эта система и являлась предметом тестирования.

Система сенсорной преднастройки реализована в виде нейронной сети преднастройки с выделением нейросетевого блока адаптивной фильтрации, входными величинами которого являются величины суставных углов Θ_1 , Θ_2 и измеряемый вес объекта манипулирования P . Выходными величинами являлись прогнозируемые значения показаний датчика силы F_1 и F_2 в условиях произвольного движения манипулятора без взаимодействия с внешними объектами.

Найденная зависимость $F_i = \Psi(\Theta_1, \Theta_2, P)$, при которых

$$\Sigma \text{mod}[F_i(\text{прогн.}) - F_i(\text{измер.})] < \varepsilon,$$

где ε — требуемая точность выделения внешнего силового воздействия, означает, что нейронная сеть адаптивной фильтрации построила модель объекта управления в условиях его свободного произвольного движения во внешней среде. После того как это условие выполняется, поиск преднастраивающих воздействий прекращается и считается, что в системе управления реализован механизм константности восприятия. Теперь любое расхождение измеряемых сенсорами и прогнозируемых системой сенсорной преднастройки значений силы расценивается системой как действие

внешнего фактора, на который нужно реагировать в соответствии с целями управления.

На рис. 10 представлены измеряемые значения датчиков силы в условиях движения манипулятора по сложной траектории. В процессе этого движения на объект в схвате производилось четыре кратковременных воздействия.

Из осциллограмм видно, что по форме и характеристикам только сенсорного сигнала определить величину, направленность и силу этих воздействий невозможно. При наличии же сенсорной преднастройки внешний силовой фактор четко выделяется. Следует отметить, что если по каким-то причинам модель объекта управления изменилась и прогнозируемые величины уже не соответствуют реально измеряемым, то может быть включен режим дообучения и модель объекта управления в нейронной сети сенсорной преднастройки будет скорректирована.

Заключение

Представленная модель адаптивной нейронной сети с преднастройкой разрабатывалась под задачи использования ее в информационно-управляющих системах роботов. Пожалуй, это наиболее перспективная область применения нейронных сетей, так как именно в этой области – управление поведением в сложной динамически меняющейся среде, естественные нейронные сети демонстрируют максимальную эффективность, а реально используемые технические решения – максимальную примитивность.

Как любое новое направление, изложенная модель вызывает больше вопросов, чем ответов. Модель должна содержать две сети: сеть, которую преднастраивают, и сеть, которая преднастраивает. Если предложенная модель преднастраиваемой нейронной сети – сети адаптивной фильтрации, может считаться адекватной и задаче и средствам, то вторая сеть – сеть преднастройки или сеть состояний, прописана более чем конспективно и заменена некоторой поисковой вычислительной процедурой.

Второй, не менее важный вопрос, связан с отображением в нейронной сети при построении модели объекта управления его динамики. Например, при резкой остановке манипулятора будут наблюдаться переходные процессы с некоторым затуханием во времени, причем характеристики этих переходных процессов зависят от параметров объекта управления (массы, люфты, выбег, гибкость звеньев и т. д.) Как все это отображать в нейрон-

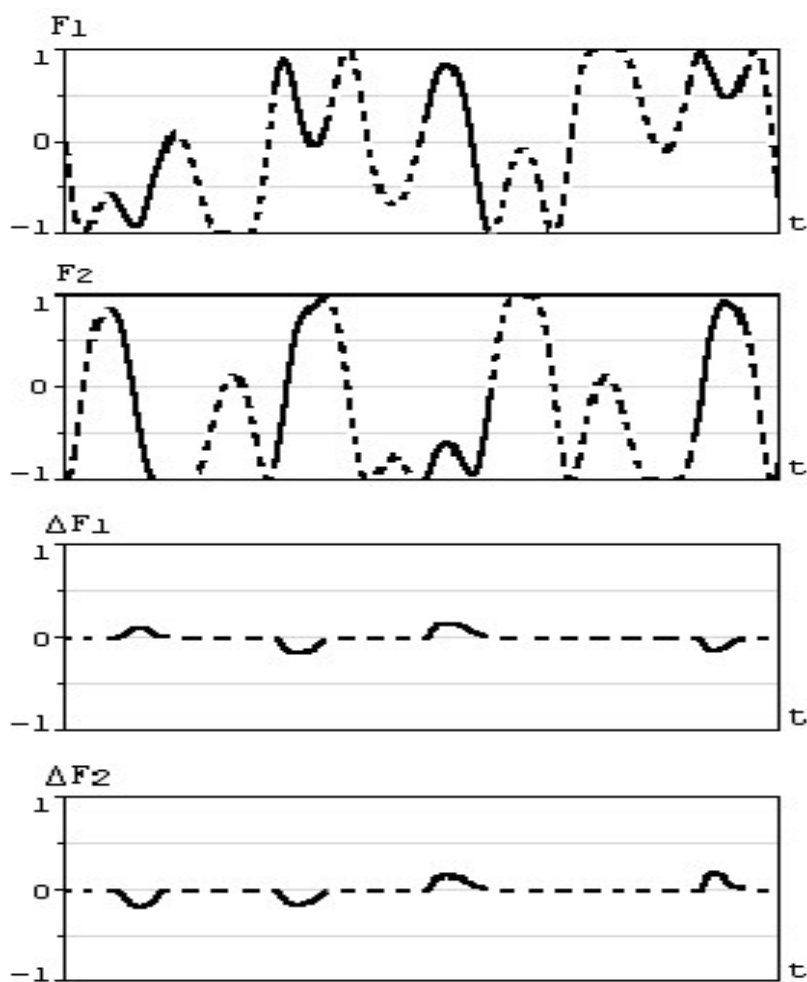


Рис. 10. Выделение внешних силовых возмущений. Показания силового датчика отражены на верхних двух графиках. Периоды действия возмущений отмечены сплошной линией. Выделенные амплитуды силовых возмущений показаны на двух нижних графиках.

ной сети?

Именно эти две проблемы — нейронная сеть преднастройки и отображение динамики объекта в активности нейронной сети являются предметом планируемых исследований.

Литература

1. *Розенблатт Ф.* Принципы нейродинамики. – М.: Мир, 1965. – 480 с.
2. *Галушкин А. И.* Теория нейронных сетей. – М.: ИПРЖР. – Сер. «Нейрокомпьютеры и их применение». Кн. 1. – 2000.
3. *Hopfield J. J.* Neural networks and physical systems with emergent collective computational abilities // *Proc. of the National Academy of Sciences.* – 1982. – v. 79. – pp. 2554–2558.
4. *Амосов Н. М.* Алгоритмы разума. – Киев: Наукова думка, 1979. – 223 с.
5. *Амосов Н. М., Касаткин А. М., Касаткина Л. М., Талаев С. А.* Автоматы и разумное поведение: Опыт моделирования. – Киев: Наукова думка, 1973. – 357 с.
6. *Байдук Т. Н.* Нейронные сети и задачи искусственного интеллекта. – Киев: Наукова думка, 2001. – 263с.
7. *Carpenter G. A., Grossberg S., Markuzon N., Reynolds J. H., Rosen D. B.* Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps // *IEEE Trans. on Neural Networks.* – 1992. – v. 3, No. 5. – pp. 698–714.
8. *Ивахненко А. Г.* Теория самоорганизации — основа прямого моделирования сложных систем по экспериментальным данным // В кн.: *Проблемы адаптивного управления.* – НТОРЭС им. А. С. Попова. – Ростов-на-Дону, 1974. – с.192–210.
9. *Анохин П. К.* Биология и нейрофизиология условного рефлекса. – М.: Медицина, 1968.
10. *Найссер У.* Познание и реальность. Смысл и принципы когнитивной психологии. – М.: Прогресс, 1981. – 230 с.
11. *Линдсей П., Норман Д.* Переработка информации у человека. – М.: Мир, 1974. – 550с.
12. *Евреинов Э. В., Прангшвили И. В.* Цифровые автоматы с настраиваемой структурой. – М.: Энергия, 1974. – 240с.
13. *Каляев А. В.* Многопроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.

14. Мак-Каллок У. С., Питтс В. Логическое исчисление идей, относящихся к нервной активности // В кн.: *Автоматы*. – М.: ИЛ, 1956. – с. 362–384.
15. Hebb D. O. *The Organization of Behavior*. – New York: Wiley, 1949.
16. McCulloch W. S. A gathe Tyche of nervous nets – the lucky reckoners // *Proc. Symposium on Mechanization of Thought Prozesses*, N. P. L., Teddington, 1958.
17. Блюм М. Свойства нейронов со многими входами / В кн.: *Принципы самоорганизации*. – М.: Мир, 1966. – с. 136–162.
18. Мак-Каллок У. Символическое изображение нейрона в виде некоторой логической функции // В кн.: *Принципы самоорганизации*. – М.: Мир, 1966. – с. 131–135.
19. Мак-Каллок У. Надежность биологических стистем // В кн.: *Самоорганизующиеся системы*. – М.: Мир, 1964. – с. 358–378.
20. Виноград С., Коуэн Дж. Д. Надежные вычисления при наличии шумов. – М.: Наука, 1968. – 112 с.
21. Мак-Каллок У. Нервные сети, устойчивые к шуму // В кн.: *Концепция информации и биологические системы*. – М.: Мир, 1966. – с. 265–276.
22. Самарин А. И., Соколенко П. Т. Функциональная универсальность и надежность нейронных сетей // В сб.: *Кибернетика и теория организации*. – НТОРЭС им. А. С. Попова. Ростов-на-Дону, 1972. – с. 124–126.
23. Самарин А. И., Финкельштейн Б. А. Синтез сетей, реализующих временные булевы функции // В сб.: *Кибернетика и теория организации*. – НТОРЭС им. А. С. Попова. Ростов-на-Дону, 1972. – с. 121–124.
24. Самарин А. И. Формирование сенсомоторных отношений при активном взаимодействии автономной системы с внешней средой. // В сб. научных трудов Всероссийской научно-технической конференции «Нейроинформатика–99». – Ч. 2. – М.: МИФИ, 1999. – с. 172–180.
25. Горбань А. Н., Россиев Д. А. Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996. – 276 с.
26. Гаврилей Ю. К., Самарин А. И., Шепелев И. Е. Преднастройка нейронных сетей в задачах зрительного анализа. // Докл. 2-ой Всероссийской научно-технической конференции «Нейроинформатика–2000». – М.: МИФИ, 2000. – с. 161–166.
27. Миркес Е. М. Функциональные модели универсального нейрокомпьютера. Дисс. на соискание ученой степени докт. техн. наук. – Красноярск, 2000.
28. Bezdek J. Fuzzy models – what are they, and why? // *IEEE Transactions on Fuzzy Systems*. – 1993, **1**(1), pp. 1–6.
29. Растрвигин Л. А. Системы экстремального управления. – М.: Наука, 1974. – 632 с.

30. *Zadeh L. A. A fuzzy logic and its application to approximate reasoning // Information Processing.* – 1974. – Vol. 3.
31. *Шенелев И. Е. Модель нейронной сети с преднастройкой для решения задач формирования сенсомоторной координации робота-манипулятора. Автореферат дисс. на соискание ученой степени канд. техн. наук. Ростов-на-Дону, 2004.* – 22 с.

Анатолий Иванович САМАРИН — заведующий отделом моделирования нервных механизмов и робототехники НИИ нейрокибернетики им. А. Б. Когана Ростовского университета. Работу в области нейронных наук начал в 1965 году с изучения свойств функциональной универсальности нейронных сетей. Практически все последующие исследования в этой области носили прикладной характер: от задач адаптивного управления с помощью нейронных сетей, до задач создания систем технического зрения и адаптивного управления поведением мобильного робота. По тематике исследования опубликовано более 120 работ. Работа по созданию системы «Глаз–рука» в 1982 году отмечена золотой медалью ВДНХ. Серия работ по созданию комплексной системы идентификации личности погибших при массовых катастрофах с использованием нейросетевых технологий отмечена правительственной наградой. Главная мечта всей жизни — познать «как мозг делает мысль». Путь для воплощения этой мечты автор видит в реализации постулата: «Только создав робота, познаешь, как работает мозг».

В. А. ГОЛОВКО

Брестский государственный технический университет

E-mail: gva@bstu.by

НЕЙРОСЕТЕВЫЕ МЕТОДЫ ОБРАБОТКИ ХАОТИЧЕСКИХ ПРОЦЕССОВ

Аннотация

В данной лекции рассмотрены нейросетевые методы для обработки хаотических процессов. Изложены стандартные методы для анализа хаотических процессов с целью определения параметров вложения динамической системы и идентификации хаоса. Рассмотрен нейросетевой метод для вычисления наибольшего показателя Ляпунова, который базируется на применении многослойного персептрона для вычисления расхождения двух близлежащих траекторий. Данный метод позволяет вычислять искомый показатель при малом объеме исходной временной последовательности и обладает меньшей вычислительной сложностью по сравнению с традиционными методами. Предложен нейросетевой метод для определения спектра Ляпунова, который основывается на применении многослойного персептрона для прогнозирования следующего состояния динамической системы на основе ее предыдущего состояния. Достоинством метода являются возможность определения показателей Ляпунова на основе наблюдаемых реализаций хаотического процесса. Исследовано применение нейронных сетей для построения странных аттракторов на основе прогнозируемых значений хаотического процесса. В качестве исходных данных использовались последовательности Энона и Лоренца. Показано, что при помощи нейронной сети можно осуществить реконструкцию динамики системы, используя малый объем исходных данных. Рассмотрен метод переобучения нейронной сети для увеличения горизонта прогнозирования хаотических процессов. Разработана нейросетевая методика обработки хаотических процессов, которая позволяет идентифицировать хаотическое поведение динамической системы, прогнозировать ее временные характеристики и реконструировать аттрактор. В отличие от известных, она позволяет использовать малый объем исходных данных хаотического процесса.

V. A. GOLOVKO

Brest State Technical University

E-mail: gva@bstu.by**NEURAL NETWORK TECHNIQUES FOR CHAOTIC SIGNAL
PROCESSING****Abstract**

We discuss in this Lecture possible using of neural network techniques for chaotic signal processing. Among discussed topics are time series analysis, identification of chaotic behavior, forecasting, and dynamic reconstruction. The fundamental aspects of chaotic time series processing are considered, namely determination of embedding parameters, Lyapunov spectrum, forecasting of chaotic data at the level of individual data points as well as emergent structure. Both conventional and neural network approaches are analyzed for chaotic signal processing. In various domains neural networks have been shown as powerful tool in comparison with conventional techniques. The neural net approach allow to evaluate the Lyapunov spectrum and to reconstruct a state space accurately and efficiently only by use of some observed data. Besides, the largest Lyapunov exponent and the Lyapunov spectrum can be computed by means of neural networks even on small data sets; this results allow both to reduce a computational complexity and to limit an observation time.

Введение

Хаотическое поведение характеризуется высокой чувствительностью к начальным условиям и наблюдается во многих областях: физике, химии, экологии, медицине и экономике [1–4]. Теория хаоса позволила объяснить многие явления, например, химические часы [2] и в настоящее время широко используется даже в социальных науках [2].

Основной характеристикой хаотических процессов является их непредсказуемость на длительный промежуток времени, так как ошибка прогнозирования увеличивается с течением времени по экспоненте. Поэтому небольшая неточность в определении начального состояния системы быстро возрастает с течением времени, что приводит к независимости следующих состояний системы от предыдущих. В результате малые причины могут привести к большим следствиям [1–4].

В теории хаоса такое явление часто называют эффектом бабочки, когда взмах крыльев бабочки в хаотической системе может вызвать со временем бурю [1]. Хаотическое поведение системы характеризуется странными аттракторами, которые имеют фрактальную размерность и обладают свойством масштабной инвариантности [4].

Одним из основателей теории хаоса является лауреат Нобелевской премии *Илья Пригожин*, который впервые объяснил многие нелинейные явления [2] и возлагал большие надежды на нейросетевые технологии [5].

Обработку хаотических процессов можно разделить на ряд этапов, как показано на рис. 1. Первый этап — это анализ временного ряда. В результате его выполнения определяются размерность пространства вложения и временная задержка сигнала. Определение параметров вложения обеспечивает максимальную предсказуемость временного ряда [6–8] и может быть использовано для выбора оптимального размера окна (количество входных элементов) в прогнозирующей нейронной сети. Используя результаты предыдущего этапа, можно осуществить идентификацию хаотического процесса путем вычисления максимального показателя Ляпунова, а также осуществить прогнозирование и фазовую реконструкцию хаотического процесса. Основной проблемой здесь является обработка хаотических процессов по наблюдаемой реализации [1, 9, 10]. В этом случае при малом объеме исходных данных проблематично использовать стандартные аналитические подходы, как для реконструкции аттрактора, так и для вычисления спектра Ляпунова.

Поэтому альтернативой здесь является разработка нейросетевых методов для обработки хаотических процессов, что, как будет показано ниже, позволяет значительно снизить трудоемкость вычислений.

В данной лекции рассматриваются как стандартные, так и нейросетевые методы обработки хаотических сигналов. Приводится нейросетевой подход для вычисления максимального показателя Ляпунова, спектра Ляпунова, прогнозирования и реконструкции фазового пространства.

Методы анализа хаотических процессов

Основные определения

Рассмотрим некоторые основные понятия теории хаоса. Под *динамической системой* обычно понимается система произвольной природы (физи-

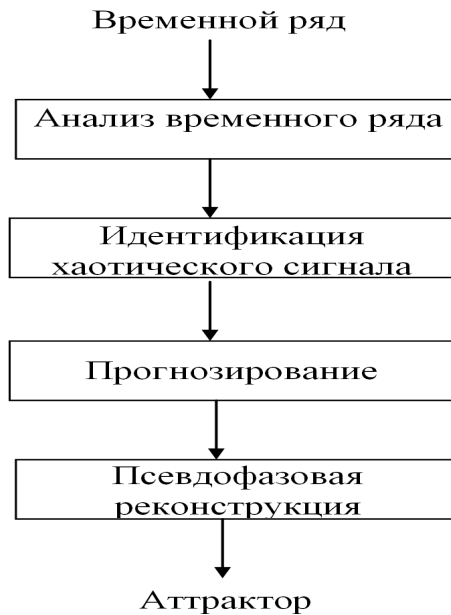


Рис. 1. Функциональная диаграмма обработки хаотических сигналов

ческой, химической, биологической и т. д.), которая может быть описана в какой-либо математической форме: в форме системы дифференциальных уравнений (обыкновенных и/или в частных производных), в форме отображений пространства. Изменение состояния динамической системы с течением времени принято называть *эволюцией* динамической системы, а уравнения, описывающие эти изменения — *уравнениями эволюции* системы. Величины, присутствующие в уравнениях эволюции и описывающие состояние динамической системы, называют *фазовыми координатами*. Множество значений фазовых координат или, говоря другими словами, пространство, в котором «живет» динамическая система называют *фазовым пространством*.

Большое влияние на эволюцию динамической системы оказывает наличие или отсутствие потерь внутренней энергии. Например, в случае если речь идет о динамической системе механической природы, то уменьшение внутренней энергии может быть связано с трением. Геометрически поте-

ря внутренней энергии означает уменьшение с течением времени объема (меры в фазовом пространстве) объекта в результате эволюции системы. Динамические системы, в которых наблюдается постоянное уменьшение внутренней энергии с течением времени, называют *диссипативными*, а в которых внутренняя энергия не изменяется — *консервативными* или *гамильтоновыми*.

Основным объектом исследования теории хаоса являются хаотические системы. *Хаотической* называется динамическая система, которая имеет долговременное аperiodическое поведение и демонстрирует чувствительность к значениям начальных условий. Под долговременным аperiodическим поведением следует понимать отсутствие периодических и квазипериодических орбит, а также орбит, сходящихся к неподвижным точкам. Чувствительность к значениям начальных условий означает, что две несовпадающие, но очень близкие точки фазового пространства имеют траектории, которые с течением времени существенно отдаляются друг от друга [11]. В результате малые причины могут привести к большим последствиям. Часто, описывая теорию хаоса, в качестве примера приводят так называемый эффект бабочки, когда взмах крыльев бабочки в хаотической системе может вызвать со временем бурю.

В качестве меры скорости расхождения близких траекторий используют *экспоненту Ляпунова (старший показатель Ляпунова)* [12, 13]. Наличие у динамической системы положительной экспоненты Ляпунова является одним из признаков хаотичности этой системы. Если рассматривать расхождение траекторий по каждому из направлений, заданных базисом фазового пространства, то можно говорить о нескольких показателях Ляпунова, образующих *спектр Ляпунова*. Более точно эти понятия будут определены ниже.

В основном хаотические системы являются диссипативными системами, однако есть примеры консервативных хаотических систем.

Любая диссипативная динамическая система, имеет *аттрактор* — компактное подмножество фазового пространства, к которому асимптотически «притягиваются» траектории эволюции всех точек системы, расположенных недалеко от этого подмножества [13]. Однако если в качестве аттрактора нехаотической диссипативной системы обычно выступает простое множество с целой размерностью, то аттрактор хаотической системы представляет собой достаточно сложный геометрический объект, имеющий нецелую фрактальную размерность. В связи с этим аттракторы хаотических систем обычно называют *странными аттракторами*. Понятие *фрактальной раз-*

мерности является обобщением обычной размерности, позволяющее более тонко отразить внутреннюю структуру объекта. Формальное определение фрактальной размерности будет приведено далее.

Основная задача анализа хаотических сигналов состоит в нахождении таких параметров системы как *размерность пространства вложения* и *временная задержка сигнала*.

Далее в лекции рассматриваются методы, позволяющие определить параметры вложения системы.

Анализ временного ряда

Пусть имеется некоторая динамическая система, но мы можем наблюдать (измерять) только одну из фазовых координат этой системы. Запишем равноотстоящие по времени измерения этой координаты в виде временного ряда:

$$x(1), x(2), \dots, x(N), \quad (1)$$

где N — количество измерений.

Основная идея оценки размерности аттрактора исходной хаотической системы основана на использовании для этих целей аттрактора, полученного при помощи так называемой псевдофазовой реконструкции.

Псевдофазовая реконструкция — это отображение, которое точке $x(t)$ временного ряда ставит в соответствие точку $[x(t), x(t + \tau), \dots, x(t + (m - 1) \cdot \tau)] \in R^m$, где t — дискретное время ($t = ((m - 1)\tau + 1), N$), τ — временная задержка (в дискретах времени) и m — размерность пространства вложения.

Такенс [13] показал, что используя только одну координату динамической системы, можно реконструировать исходный аттрактор в пространстве точек с задержками $[x(t), x(t + \tau), \dots, x(t + (m - 1) \cdot \tau)]$ таким образом, что он будет сохранять важнейшие топологические свойства и динамику оригинального аттрактора. Размерность m определяется по формуле $m \geq 2[d] + 1$, где d — фрактальная размерность аттрактора, $[d]$ — обозначает целую часть фрактальной размерности.

Таким образом, анализ хаотических процессов заключается в определении параметров вложения динамической системы, а именно в выборе подходящей временной задержки сигнала τ [11, 18–20] и размерности m [14–17, 21–24] пространства вложения для псевдофазовой реконструкции.

Данные параметры необходимы для максимальной предсказуемости хаотического процесса и, как будет показано далее, для выбора структуры нейронной сети.

Выбор временной задержки

Для выбора временной задержки τ могут быть использованы следующие подходы [11, 18–20]:

- метод автокорреляционной функции;
- метод взаимной информации;
- метод среднего отклонения.

Первые два метода основаны на предположении, что временная задержка является оптимальной, если координаты реконструированного аттрактора являются максимально независимыми притом, что сама задержка τ имеет как можно меньшее значение.

Метод автокорреляционной функции. Рассмотрим *автокорреляционную функцию*:

$$R(\tau) = \frac{1}{N_1} \sum_{t=1}^{N_1} y(t)y(t + \tau), \quad (2)$$

где $N_1 = N - \tau$, $y(t) = x(t) - E\{x\}$ — центрированная версия ряда (1).

Для каждого конкретного значения τ функция $R(\tau)$ возвращает коэффициент корреляции между исходным временным рядом (1) и его версией, полученной при помощи временной задержки на τ шагов.

Оптимальная временная задержка обычно выбирается в соответствии с первым нулевым (либо близким к нулю) значением автокорреляционной функции. Данный метод является достаточно простым и не требует больших вычислений. Но поскольку вместо свойства независимости используется неэквивалентное ему свойство некоррелированности, то полученное значение задержки τ не всегда является оптимальным.

Метод взаимной информации. Пусть $(a, b) \in R^1$ — минимальный интервал, содержащий все значения временного ряда (1). Разобьем данный интервал на L равных частей (количество интервалов разбиения обычно выбирают по известной формуле Старка $L = (\lceil \log_2 N \rceil + 1)$). Обозначим

событие «значение $x(t)$ принадлежит i -му интервалу» через A_i , а событие «значение $x(t + \tau)$ принадлежит j -му интервалу» через B_j . Тогда *функция взаимной информации* определяется соотношением:

$$I(\tau) = - \sum_{i=1}^L \sum_{j=1}^L P(A_i B_j) \cdot \log_2 \frac{P(A_i B_j)}{P(A_i)P(B_j)}, \quad (3)$$

где $P(\cdot)$ — вероятность соответствующего события.

Функция взаимной информации является, с одной стороны, более точной мерой независимости, чем автокорреляционная функция, а с другой стороны — более сложной в вычислительном смысле. В данном методе оптимальная задержка выбирается в соответствии с первым минимумом функции $I(\tau)$.

Для ускорения расчетов можно предложить модифицированную версию формулы (3):

$$I_1(\tau) = - \sum_{i=1}^L \sum_{j=1}^L [P(A_i B_j) - P(A_i)P(B_j)]^2. \quad (4)$$

Функция $I_1(\tau)$ «ведет себя» так же, как $I(\tau)$, то есть обе эти функции возрастают и убывают на одних и тех же промежутках, а также достигают экстремумов в одних и тех же точках. Расчеты показывают, что использование формулы (4) вместо (3) позволяет сократить время вычислений на 10–15%.

Как видно из рис. 2, для первой координаты системы Лоренца первый минимум функций взаимной информации и модифицированной взаимной информации достигается при $\tau = 16$ (0.16 с), а первое ближайшее к нулю значение автокорреляционной функции достигается при $\tau = 47$ (0.47 с).

Метод среднего отклонения. Данный метод был впервые предложен Розенштейном [18] в 1994 году. Он основан на оценке среднего отклонения точек реконструированного аттрактора от главной диагонали пространства вложения. Рассмотрим *функцию среднего отклонения*, заданную соотношением:

$$S_m(\tau) = \frac{1}{N_1} \sum_{t=1}^{N_1} \sqrt{\sum_{k=1}^{m-1} [x(t + k\tau) - x(t)]^2}, \quad (5)$$

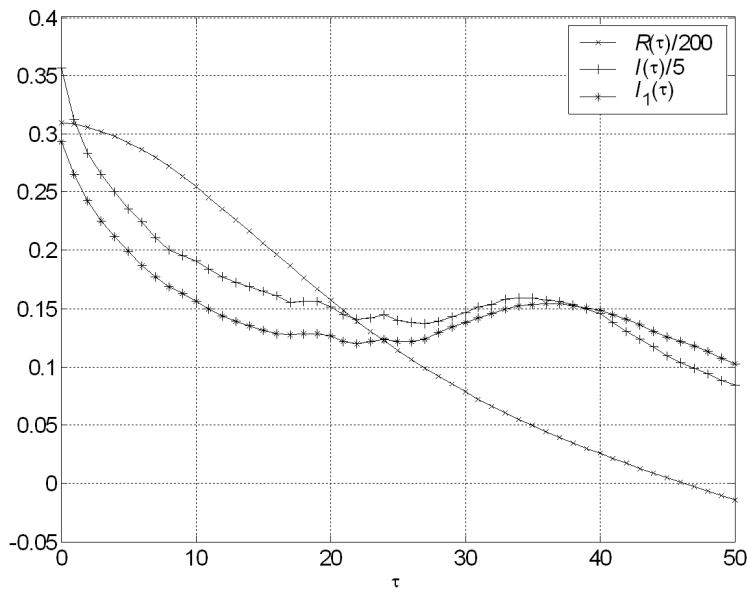


Рис. 2. Графики автокорреляционной функции $R(\tau)$, взаимной информации $I(\tau)$ и модифицированной взаимной информации $I_1(\tau)$ первой координаты системы Лоренца дискретизированной с шагом 0.01 (τ измеряется в дискретах времени). Для большей наглядности графики функций $R(\tau)$ и $I(\tau)$ изображены в измененном вертикальном масштабе.

где τ — временная задержка (в шагах), m — размерность пространства вложения, а $N_1 = N - m\tau$.

Очевидно, что при $\tau = 0$ реконструированная траектория представляет собой некоторое подмножество главной диагонали пространства вложения. При увеличении временной задержки τ реконструированная траектория постепенно отдаляется от главной диагонали, а функция S_m возрастает и достигает насыщения. Значение τ , при котором начинается насыщение, считается оптимальным. Для больших значений m , функция $S_m(\tau)$ достигает насыщения при меньших τ . Как можно заметить, данный метод наиболее эффективен, когда размерность вложения m известна априори.

Выбор размерности пространства вложения

Следующий шаг — выбор размерности m пространства вложения [14–17, 21–24]. Рассмотрим следующие методы оценки m :

- метод корреляционной размерности;
- метод ложных ближайших соседей;
- гамма-тест.

Метод корреляционной размерности. Рассмотрим упоминавшееся выше понятие *фрактальной размерности*.

Пусть в пространстве R^n задано некоторое ограниченное множество F . Зафиксируем некоторое $\varepsilon > 0$, и рассмотрим конечное покрытие $\{E_i\}_i$ этого множества. Каждый элемент покрытия E_i — замкнутый n -мерный куб, имеющий ребро длиной ε , и каждая координата каждой вершины имеет вид $k\varepsilon$, $k \in \mathbf{Z}$. Обозначив количество элементов покрытия через $N(\varepsilon)$, можно заметить, что между $N(\varepsilon)$ и ε существует зависимость вида:

$$N(\varepsilon) \cong \varepsilon^{-D_0}. \quad (6)$$

Отсюда можно выразить D_0 :

$$D_0 = - \lim_{\varepsilon \rightarrow 0} \log_{\varepsilon} N(\varepsilon). \quad (7)$$

Величину, определенную таким образом, называют *емкостью множества* F . Наряду с емкостью D_0 вводят еще ряд других размерностей (*размерность Хаусдорфа* d_H , *информационную размерность* D_1 , *корреляционную размерность* D_2 и т. д.), но для большинства задач нелинейной динамики и теории хаоса эти размерности совпадают. Поэтому терминологически их часто не различают и говорят просто о *фрактальной размерности*, обозначая эту величину через d .

Для оценки фрактальной размерности используют *алгоритм Грассберга-Прокаччия*, позволяющий вычислить корреляционную размерность D_2 множества F по выборке точек $\{\mathbf{x}_i\}_{i=1, \overline{M}}$ из этого множества.

Пусть множество F покрыто ячейками одинакового размера ε , а вероятность того, что выбранная наугад точка выборки попадет в i -й элемент покрытия, равна p_i . Тогда корреляционная размерность определяется соотношением:

$$D_2 = \lim_{\varepsilon \rightarrow 0} \frac{\ln \sum_{i=1}^{N(\varepsilon)} p_i^2}{\ln \varepsilon} = \lim_{\varepsilon \rightarrow 0} \frac{\ln C(\varepsilon)}{\ln \varepsilon}. \quad (8)$$

Величину $C(\varepsilon)$ называют *корреляционным интегралом* и вычисляют так:

$$C(\varepsilon) = \sum_{i=1}^{N(\varepsilon)} p_i^2 = \lim_{M \rightarrow \infty} \frac{1}{M(M-1)} \sum_{i=1}^M \sum_{j=1}^M \theta(\varepsilon - \|\mathbf{x}_i - \mathbf{x}_j\|), \quad (9)$$

где $\theta(\cdot)$ — функция Хевисайда,

$$\theta(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases} \quad (10)$$

Алгоритм Грассберга-Прокачиа состоит в следующем: необходимо вычислить корреляционный интеграл $C(\varepsilon)$ для разных ε , отложить полученную зависимость в системе координат с двойным логарифмическим масштабом, отыскать на ней линейный участок и найти угловой коэффициент d этого участка. Полученная величина d и будет оценкой корреляционной размерности множества F .

Рассмотрим, как при помощи приведенного выше алгоритма можно найти размерность пространства вложения m . Пусть имеется ряд (1) и вычисленная каким-либо способом временная задержка τ . Тогда зафиксировав целое m и применив идею псевдофазовой реконструкции, можно получить набор точек:

$$\mathbf{x}_i = [x(i), x(i - \tau), \dots, x(i - (m - 1) \cdot \tau)] \in \mathbf{R}^m, \quad (11)$$

где $i = \overline{[(m - 1)\tau + 1], N}$. Для данного числа m и полученной выборки \mathbf{x}_i можно описанным выше методом найти корреляционную размерность $D(m)$.

Описанная процедура выполняется несколько раз, рассматривая последовательно $m = 1, 2, 3, \dots$. При увеличении значения m наблюдается насыщение соответствующей величины $D(m)$. Если насыщения $D(m)$ нет, то рассматриваемый сигнал, скорее всего, генерируется не динамической системой, а является шумом. Если наблюдается насыщение $D(m)$ на уровне d , то величина d принимается в качестве оценки корреляционной размерности, а величина m с которой начинается насыщение $D(m)$, принимается в качестве оценки размерности пространства вложения.

Метод ложных ближайших соседей. Метод ложных ближайших соседей [23] основан на теореме Такенса о вложении, из которой следует, что

при соответствующем выборе τ и m оригинальный и реконструированный аттракторы должны быть топологически эквивалентны (гомеоморфны). Поскольку траектории оригинального аттрактора не имеют самопересечений, то и в реконструированном аттракторе траектории также не должны пересекаться. Самопересечение траекторий реконструированного аттрактора означает, что размерность m пространства вложения меньше фрактальной размерности аттрактора, то есть соответствующая псевдофазовая реконструкция не является биекцией. Условием того, что самопересечения будут отсутствовать, является то, что все соседние точки аттрактора восстановленного в R^m , будут также являться соседними в R^{m+1} . Метод ложных ближайших соседей позволяет определить наименьшее значение размерности m пространства вложения, так что при переходе к размерности $(m+1)$ количество ложных соседей (точек аттрактора, близких друг к другу в R^m и отстоящих далеко в R^{m+1}) будет относительно мало. Полученное таким образом значение m определяет наименьшую размерность пространства вложения, где возможна реконструкция аттрактора без самопересечений.

Алгоритм метода ложных ближайших соседей состоит из следующих шагов:

1. Пусть $m = 1$. Находим для каждой точки $\bar{x}(i)$ временного ряда ближайшего «соседа» $\bar{x}(j)$ в m -мерном пространстве.
2. Вычисляем расстояние $\|\bar{x}(i) - \bar{x}(j)\|$.
3. Находим расстояние между данными точками на следующем шаге $\|\bar{x}(i+1) - \bar{x}(j+1)\|$ и определяем

$$R_i = \frac{\|\bar{x}(i+1) - \bar{x}(j+1)\|}{\|\bar{x}(i) - \bar{x}(j)\|}.$$

4. Если $R_i > R_t$, где R_t — подходящий порог, то точка $\bar{x}(j)$ является ложным ближним соседом по отношению к точке $\bar{x}(i)$. В результате подсчитывается количество таких ложных ближних соседей P для каждой точки $\bar{x}(i)$.
5. Вычисляется P/N и алгоритм повторяется для $m = m + 1$.
6. Алгоритм продолжается до тех пор, пока частное P/N не станет близким к нулю.

Рекомендуемое значение $R_t = 2$.

На рис. 3 и 4 представлены графики для определения размерности пространства вложения для данных Энона и Лоренца соответственно. Из ри-

сунков определяется размерность, равная трем, для данных Энона и пяти для данных Лоренца.

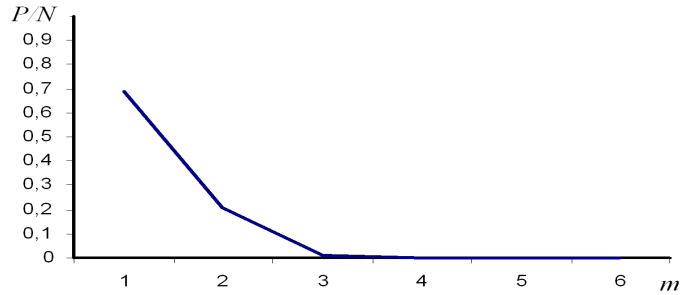


Рис. 3. Определение размерности пространства вложения для аттрактора Энона

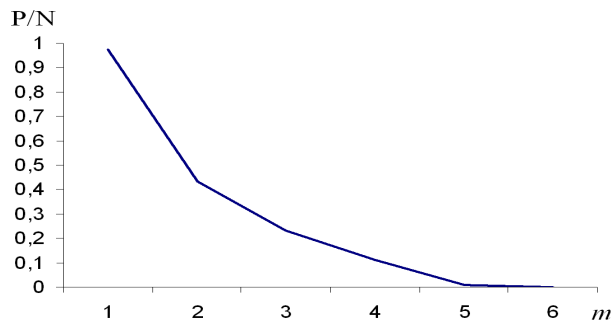


Рис. 4. Определение размерности пространства вложения для аттрактора Лоренца

Гамма-тест. Пусть имеется непрерывное отображение $f : \mathbf{R}^m \rightarrow \mathbf{R}^1$ и пусть y определяется соотношением:

$$y = f(x_1, x_2, \dots, x_m) + r, \quad (12)$$

где r — неопределенная составляющая, которая может иметь место, например, благодаря шуму. Основное назначение гамма-теста — оценить дисперсию величины r , то есть нижнюю границу квадратичной ошибки вычисления величины y .

Пусть дана выборка $\mathbf{X} = \{\mathbf{x}_i \in \mathbf{R}^m, i = \overline{1, M}\}$ m -мерных векторов; каждому вектору \mathbf{x}_i в соответствии с соотношением (12) сопоставим величину y_i . Для каждого вектора \mathbf{x}_i обозначим через \mathbf{x}_{i_p} p -го ближайшего соседа из выборки \mathbf{X} в метрике пространства \mathbf{R}^m , а через y_{i_p} — величину, соответствующую по (12) вектору \mathbf{x}_{i_p} . Рассмотрим следующие функции:

$$\Delta(p) = \frac{1}{p} \sum_{h=1}^p \frac{1}{M} \sum_{i=1}^M \|\mathbf{x}_i - \mathbf{x}_{i_h}\|^2 \quad (13)$$

и

$$\Gamma(p) = \frac{1}{p} \sum_{h=1}^p \frac{1}{2M} \sum_{i=1}^M (y_i - y_{i_h})^2. \quad (14)$$

Для каждого $p = \overline{1, p_{\max}}$ (p_{\max} обычно берется в пределах от 20 до 50) можно вычислить пару значений $\langle \Delta(p), \Gamma(p) \rangle$, для которых можно выявить линейную зависимость в виде уравнения линейной регрессии $y = Ax + \bar{\Gamma}$ и методом наименьших квадратов оценить коэффициенты A и $\bar{\Gamma}$. Значение коэффициента A характеризует сложность поверхности, определенной в пространстве \mathbf{R}^{m+1} соотношением (12), а значение $\bar{\Gamma}$ и есть искомое значение гамма-теста.

Рассмотрим применение гамма-теста [24] для оценки размерности вложения.

Пусть известна лишь одна фазовая координата эволюции динамической системы, дискретизированная с некоторым малым временным шагом. Обозначим значения этого временного ряда через $x(t)$, где t обозначает дискретное время. Следуя стандартной процедуре реконструкции аттрактора системы по одной координате, рассмотрим набор m -мерных векторов:

$$\mathbf{x}(t) = [x(t), x(t - \tau), x(t - 2\tau), \dots, x(t - (m - 1)\tau)], \quad (15)$$

где τ — предварительно выбранная временная задержка.

Определим отображение f , действующее из множества рассмотренных выше векторов $\mathbf{x}(t)$ в пространство \mathbf{R}^1 , соотношением:

$$f(\mathbf{x}(t)) = x(t + 1). \quad (16)$$

Для данного отображения можно применить гамма-тест, то есть оценить $\bar{\Gamma}$. Понятно, что для каждой размерности m значение $\bar{\Gamma}$ будет разным, таким образом можно говорить о зависимости $\bar{\Gamma}$ от m , то есть о функции $\bar{\Gamma} = \bar{\Gamma}(m)$. Расчеты показывают, что обычно эта функция при увеличении размерности m сначала убывает, а затем возрастает. Значение m , при котором $\bar{\Gamma}(m)$ достигает минимума и рекомендуется брать в качестве размерности вложения, то есть

$$m = \arg \min_k \{\bar{\Gamma}(k)\}. \quad (17)$$

Однако возможны ситуации, когда $\bar{\Gamma}(m)$ монотонно возрастает, либо монотонно убывает. Причинами такого поведения являются либо неудачный выбор временной задержки τ , либо большой уровень шума во входных данных.

Таким образом, в данном разделе рассмотрены методы определения параметров вложения хаотического процесса, которые необходимы для обеспечения максимальной предсказуемости хаотического процесса и используются во всех последующих вычислениях.

Определение старшего показателя Ляпунова

Как было сказано выше, одним из условий хаотичности динамической системы является *чувствительность к значениям начальных условий*. В качестве количественной меры этого понятия обычно рассматривают так называемый *старший показатель Ляпунова (экспоненту Ляпунова)* [25–32].

Понятие старшего показателя Ляпунова

Рассмотрим точку $\mathbf{x}(t_0)$, принадлежащую аттрактору некоторой динамической системы в начальный момент времени t_0 . Задавшись некоторым малым положительным числом $\varepsilon(t_0)$, выберем на аттракторе еще одну точку $\tilde{\mathbf{x}}(t_0)$ так, чтобы $\|\tilde{\mathbf{x}}(t_0) - \mathbf{x}(t_0)\| = \varepsilon(t_0)$. Через промежуток времени Δt точки $\mathbf{x}(t_0)$ и $\tilde{\mathbf{x}}(t_0)$ эволюционируют соответственно в $\mathbf{x}(t)$ и $\tilde{\mathbf{x}}(t)$, расстояние между ними обозначим через $\varepsilon(t)$, где $t = t_0 + \Delta t$ (рис. 5).

Очевидно, что $\varepsilon(t)$ зависит от положения точек $\mathbf{x}(t_0)$ и $\tilde{\mathbf{x}}(t_0)$, промежутка времени Δt и, конечно, свойств самой динамической системы. Однако,

упрощенно, можно считать, что

$$\varepsilon(t) \cong \varepsilon(t_0) e^{\lambda \Delta t}, \quad (18)$$

где λ — параметр, характеризующий динамическую систему и называемый *старшим показателем Ляпунова* или *экспонентой Ляпунова*.

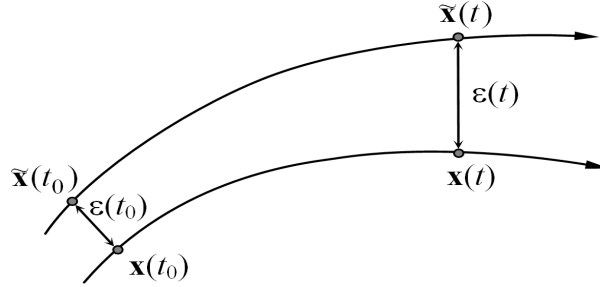


Рис. 5. К определению понятия экспоненты Ляпунова

После элементарных преобразований получим:

$$\lambda \cong \frac{1}{\Delta t} \ln \frac{\varepsilon(t)}{\varepsilon(t_0)}. \quad (19)$$

Сразу необходимо сделать два замечания:

- во-первых, в последнем соотношении, в силу ограниченности аттрактора (а значит ограниченности $\varepsilon(t)$) Δt должно возрастать до тех пор, пока $\varepsilon(t)$ существенно меньше размеров аттрактора, иначе λ будет равен нулю, при $\Delta t \rightarrow \infty$;
- во-вторых, вычисленное в соответствии с (19) значение λ следует рассматривать как усредненное по всем начальным точкам $\mathbf{x}(t_0)$ аттрактора системы.

С учетом этих замечаний можно строго определить старший показатель Ляпунова соотношением:

$$\lambda = \underset{\mathbf{x}(t_0) \in A}{E} \left\{ \lim_{\substack{\Delta t \rightarrow \infty \\ \varepsilon(t) < \text{diam} A}} \frac{1}{\Delta t} \ln \frac{\varepsilon(t)}{\varepsilon(t_0)} \right\}, \quad (20)$$

где A — аттрактор системы, $\text{diam}A$ — его диаметр, а $E\{\cdot\}$ — математическое ожидание.

Старший показатель Ляпунова характеризует степень экспоненциального расхождения близких траекторий. Наличие у системы положительной экспоненты Ляпунова свидетельствует о том, что любые две близкие траектории быстро расходятся с течением времени, то есть имеет место *чувствительность к значениям начальных условий*. Поэтому определение экспоненты Ляпунова позволяет идентифицировать динамическую систему с точки зрения присутствия в ней хаотического поведения.

Алгоритм Бенеттина

Для оценки старшего показателя Ляпунова наиболее часто используют *алгоритм Бенеттина* [25].

Пусть имеется x_0 — точка, принадлежащая аттрактору A исследуемой динамической системы. Траекторию эволюции точки x_0 будем называть *опорной траекторией*. Зададимся положительной величиной ε , удовлетворяющей условию $\varepsilon \ll \text{diam}A$ и выберем произвольным образом такую *точку возмущения* \tilde{x}_0 , чтобы выполнялось равенство $\|\tilde{x}_0 - x_0\| = \varepsilon$. Рассмотрим эволюцию выбранных точек x_0 и \tilde{x}_0 в течение небольшого интервала времени T и обозначим полученные точки через x_1 и \tilde{x}_1 , соответственно. Вектор $\Delta x_1 = \tilde{x}_1 - x_1$ назовем *вектором возмущения*, а его длину $\|\Delta x_1\|$ — *амплитудой возмущения*. Уже на данном этапе можно произвести первую оценку величины λ :

$$\tilde{\lambda}_1 = \frac{1}{T} \ln \frac{\|\Delta x_1\|}{\varepsilon}. \quad (21)$$

Временной интервал T необходимо брать таким, чтобы амплитуда возмущения была меньше линейных размеров неоднородностей фазового пространства и, тем более, размеров самого аттрактора. Рассмотрим перенормированный вектор возмущения

$$\Delta x'_1 = \frac{\Delta x_1}{\|\Delta x_1\|} \varepsilon$$

и соответствующую ему новую точку возмущения $\tilde{x}'_1 = x_1 + \Delta x'_1$. Далее, продолжим описанную процедуру, рассматривая вместо точек x_0 и \tilde{x}_0 точки x_1 и \tilde{x}'_1 , соответственно (рис. 6).

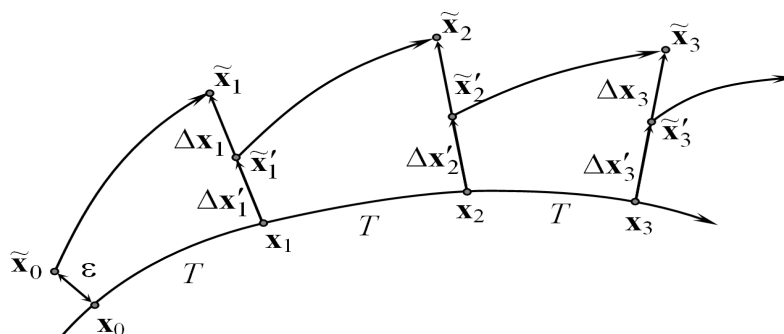


Рис. 6. К оценке экспоненты Ляпунова с помощью алгоритма Бенеттина

Повторив данную процедуру M раз, можно оценить λ как среднее арифметическое величин $\tilde{\lambda}_i$, полученных на каждом шаге вычислений:

$$\lambda \cong \frac{1}{M} \sum_{i=1}^M \tilde{\lambda}_i = \frac{1}{M} \sum_{i=1}^M \frac{1}{T} \ln \frac{\|\Delta \mathbf{x}_i\|}{\varepsilon} = \frac{1}{MT} \sum_{i=1}^M \ln \frac{\|\Delta \mathbf{x}_i\|}{\varepsilon}. \quad (22)$$

Очевидно, что для более точной оценки следует брать значение M как можно бóльшим и проводить расчеты для разных начальных точек \mathbf{x}_0 .

Оценка старшего показателя Ляпунова по выборке из одной наблюдаемой координаты

Часто возникает ситуация, когда уравнения эволюции системы неизвестны и, более того, измерить все фазовые координаты системы невозможно. В таких случаях для оценки старшего показателя Ляпунова хаотической системы используют метод, основанный на обработке измерений одной координаты рассматриваемой системы. Этот метод был впервые предложен Вольфом [26] и основан на теореме Такенса [13] и уже рассмотренном алгоритме Бенеттина.

Как уже отмечалось, Такенс доказал, что система, реконструированная методом временных задержек из одной фазовой координаты системы, сохраняет важнейшие динамические и геометрические свойства исходной,

если выполняется неравенство:

$$m \geq 2[d] + 1, \quad (23)$$

где m — размерность пространства вложения, d — фрактальная размерность аттрактора исходной системы, $[\cdot]$ — целая часть. В данном утверждении под фразой *важнейшие динамические свойства* следует понимать диссипативность и хаотичность системы, а под *важнейшими геометрическими свойствами* — топологические инварианты аттрактора такие, например, как фрактальная размерность.

Выполнение неравенства (23) гарантирует сохранение указанных свойств, однако эксперименты показывают, что данная нижняя оценка размерности пространства вложения является несколько завышенной. Для некоторых простых хаотических систем сохранение упомянутых свойств наблюдается даже при $m = [d + 1]$.

Пусть имеется временной ряд $x(t)$, $t = \overline{1, N}$ измерений одной координаты хаотического процесса, произведенных через равные промежутки времени. Тогда, используя вышеописанные методы, необходимо определить размерность пространства вложения m и временную задержку τ . В результате реконструкции получим набор точек пространства \mathbf{R}^m :

$$\mathbf{x}_i = [x(i), x(i - \tau), \dots, x(i - (m - 1) \cdot \tau)], \quad (24)$$

где $i = \overline{[(m - 1)\tau + 1], N}$.

Выберем из последовательности (24) точку и обозначим ее через \mathbf{x}_0 . Просматривая последовательность (24), найдем такую точку $\tilde{\mathbf{x}}_0$, чтобы выполнялось соотношение $\|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\| = \varepsilon_0 < \varepsilon$, где ε — фиксированная величина, существенно меньшая размеров реконструированного аттрактора. Причем необходимо, чтобы точки \mathbf{x}_0 и $\tilde{\mathbf{x}}_0$ были разделены по времени. Затем отслеживаем эволюцию выбранных точек на реконструированном аттракторе до тех пор, пока расстояние между ними не превысит заданную величину ε_{max} . Обозначим полученные точки через \mathbf{x}_1 и $\tilde{\mathbf{x}}_1$, расстояние между ними через ε'_0 , а промежуток времени эволюции через T_1 .

Далее, вновь просматривая последовательность (24), находим такую точку $\tilde{\mathbf{x}}'_1$, чтобы она была близка к \mathbf{x}_1 , то есть $\|\tilde{\mathbf{x}}'_1 - \mathbf{x}_1\| = \varepsilon_1 < \varepsilon$, а векторы $\tilde{\mathbf{x}}_1 - \mathbf{x}_1$ и $\tilde{\mathbf{x}}'_1 - \mathbf{x}_1$ имели, по-возможности, одинаковое направление. Далее процедура повторяется, но вместо точек \mathbf{x}_0 и $\tilde{\mathbf{x}}_0$ рассматриваются точки \mathbf{x}_1 и $\tilde{\mathbf{x}}'_1$, соответственно (рис. 7).

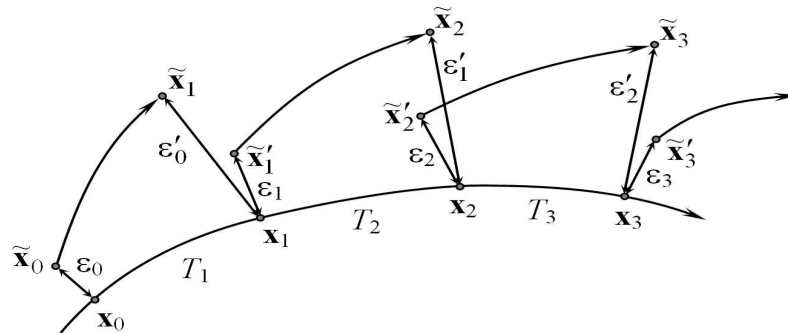


Рис. 7. К оценке старшего показателя Ляпунова по выборке из одной наблюдаемой координаты

Повторив описанную процедуру M раз, можно оценить старший показатель Ляпунова так:

$$\lambda \cong \frac{\sum_{k=0}^{M-1} \ln(\varepsilon'_k / \varepsilon_k)}{\sum_{k=1}^M T_k}. \quad (25)$$

Такой подход базируется на эргодической теореме В. И. Оселедда [16], которая утверждает, что экспоненциальное расхождение двух случайно выбранных точек на аттракторе с единичной вероятностью характеризует старший показатель Ляпунова.

Оценка старшего показателя Ляпунова по выборке из одной наблюдаемой координаты с помощью нейронной сети

Приведенный выше метод характеризуется большой вычислительной сложностью и невозможностью применения для малого объема исходных данных. Это связано с тем, что трудно найти две точки ряда, отстоящие друг от друга на расстояние меньше чем 10^{-8} . Особенно это проблематично для реальных данных. Одним из путей для преодоления этого недостатка является применение нейронных сетей для вычисления старшего показателя Ляпунова.

Ключевой идеей данного метода является вычисление при помощи прогнозирующей нейронной сети расхождения двух близлежащих траекторий на n шагов вперед [27–32].

Нейронная сеть будет состоять из $k \geq m - 1$ входных нейронов, p скрытых и одного выходного нейронного элемента (рис. 8). Здесь m – размерность пространства вложения.

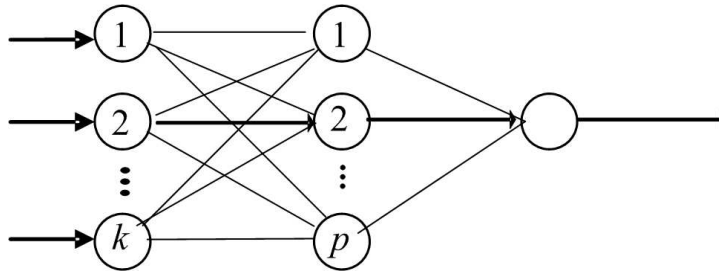


Рис. 8. Прогнозирующая нейронная сеть

Вначале необходимо обучить такую нейронную сеть прогнозированию в соответствии с методом скользящего окна:

$$x(t + i\tau) = F[x(t + (i - 1)\tau), x(t + (i - 2)\tau), \dots, x(t + (i - k)\tau)], \quad i = \overline{1, n}. \quad (26)$$

После обучения сети легко осуществить эволюцию двух точек на фазовой траектории, используя итерационный подход.

Таким образом, ключевой идеей предлагаемого метода является вычисление при помощи прогнозирующей нейронной сети расхождения двух близлежащих траекторий на n шагов вперед, используя итерационный подход. Эта процедура может быть представлена следующим алгоритмом:

1. Обучаем нейронную сеть на прогнозирование по методу скользящего окна.
2. Выбираем любую точку $x(t)$ из обучающей выборки и формируем следующий набор данных: $x(t), x(t - \tau), \dots, x(t - (k - 1)\tau)$, где k – размер окна.
3. Вычисляем $x(t + \tau), x(t + 2\tau), \dots, x(t + n\tau)$, используя многошаговый прогноз

$$x(t + i\tau) = F[x(t + (i - 1)\tau), x(t + (i - 2)\tau), \dots, x(t + (i - k)\tau)], \quad (27)$$

где $i = \overline{1, n}$, F — нелинейная функция.

4. Вычисляем $x'(t) = x(t) + d_0$, где $d_0 \approx 10^{-8}$ и, подавая на сеть $x'(t), x(t - \tau), \dots, x(t - (k - 1)\tau)$, повторяем шаг 3 для получения $x'(t + i\tau)$, $i = \overline{1, n}$.
5. Оцениваем $\ln d_i = \ln |x'(t + i\tau) - x(t + i\tau)|$, $i = \overline{1, n}$ и выбираем только точки, где $\ln d < 0$.
6. Строим график $\ln(d_n)$ от n .
7. Строим прямую регрессии для выбранных точек и вычисляем ее наклон, который равен наибольшему показателю Ляпунова.

Данный метод для вычисления λ был апробирован на многослойном персептроне. Для эксперимента использовалась нейронная сеть с 7 входными элементами, 5 скрытыми и 1 выходным нейроном для предсказания временных рядов Лоренца и Энона [1, 4]. Элементы скрытого слоя имели сигмоидную функцию активации, а выходной элемент — линейную. Для обучения сети использовался алгоритм обратного распространения ошибки с адаптивным шагом. Обучающая выборка состояла из 70 элементов для данных Энона, смоделированных через $\tau = 1$ и 100 элементов для данных Лоренца, взятых через $\tau = 0.16$ соответственно. Среднеквадратичная ошибка обучения сети для данных Энона составила $5.92 \cdot 10^{-5}$ после 1000 итераций. Рис. 9 показывает график $\ln d(n)$ от n и прямую регрессии, которая определяет показатель Ляпунова для временного ряда Энона по координате x . Оцененное значение $\hat{\lambda} = 0.43$ близко к ожидаемому значению 0.419 для данных Энона. Среднеквадратичная ошибка для ряда Лоренца составила $9.2 \cdot 10^{-4}$ после 700 итераций. Прямая регрессии и функция $\ln(d_n)$ от n для ряда Лоренца представлены на рис. 10. Наибольший показатель Ляпунова равен 0.98 (ожидаемое значение 0.906). Можно заметить, что нейронная сеть дает довольно точную оценку показателя Ляпунова при малом объеме исходных данных. Очевидное преимущество нейросетевого подхода в сравнении с традиционным — простота и точность.

Понятие спектра Ляпунова

Гораздо более тонко динамику системы отражает понятие, обобщающее понятие экспоненты Ляпунова — *спектр Ляпунова*. Спектр Ляпунова позволяет оценить скорость расхождения (сближения) траекторий вдоль разных координатных направлений.

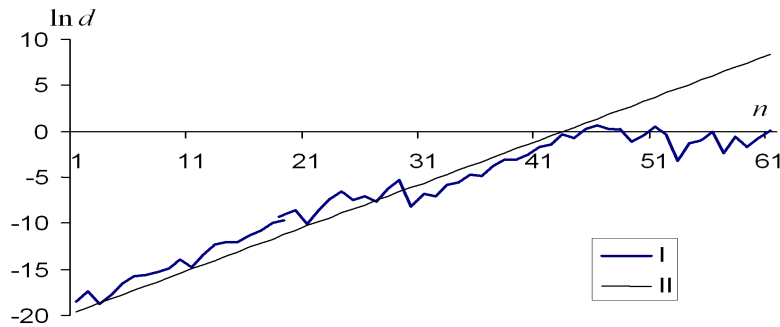


Рис. 9. Изменение расстояния между двумя близкими траекториями для аттрактора Энона (I); прямая регрессии (II)

Понятие спектра Ляпунова

Рассмотрим геометрический — наиболее простой и наглядный подход к определению понятия спектра Ляпунова [16].

Пусть дана динамическая система, имеющая n -мерное фазовое пространство. Такая система имеет n показателей Ляпунова $\lambda_i, i = 1, 2, \dots, n$, которые называются спектром Ляпунова. Зафиксируем малое положительное число ε и рассмотрим в фазовом пространстве n -мерную сферу, имеющую радиус ε . Через некоторый небольшой промежуток времени Δt данная сфера эволюционирует в n -мерный эллипсоид (рис. 11).

Обозначим главные полуоси полученного эллипсоида через $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$. Для определенности будем считать, что $\varepsilon_1 \geq \varepsilon_2 \geq \dots \geq \varepsilon_n$. Далее, аналогично введенному выше понятию экспоненты Ляпунова, можно определить показатели экспоненциального изменения полуосей эллипсоида с помощью соотношения:

$$\lambda_i \cong \frac{1}{\Delta t} \ln \frac{\varepsilon_i}{\varepsilon}, \quad (28)$$

где $i = 1, 2, \dots, n$.

Таким образом, каждый показатель Ляпунова характеризует изменение размеров эллипсоида по одной из его главных осей. Для их вычисления необходимо отслеживать изменение длин соответствующих векторов вдоль фазовой траектории.

В данном случае необходимо иметь в виду замечания, приведенные

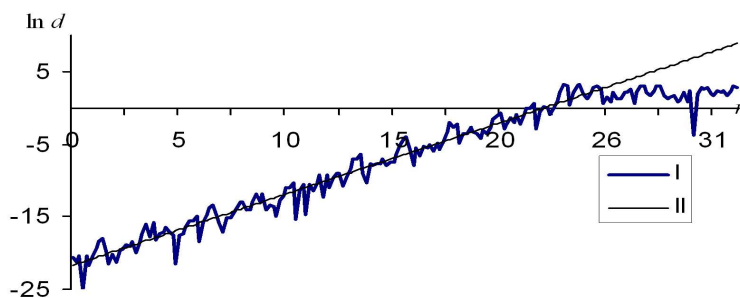


Рис. 10. Изменение расстояния между двумя близкими траекториями для аттрактора Лоренца (I); прямая регрессии (II)

выше, при определении понятия экспоненты Ляпунова. Нетрудно заметить, что наибольшее значение в спектре Ляпунова совпадает с экспонентой Ляпунова. Таким образом, определение спектра Ляпунова — задача более общая по сравнению с задачей определения экспоненты Ляпунова.

Численный расчет спектра Ляпунова

Для численного расчета спектра Ляпунова используют подход, обобщающий алгоритм Бенеттина [1]. В данном случае, кроме просчета опорной траектории, необходимо также отслеживать эволюцию не одной, а нескольких возмущенных точек. Количество просчитываемых возмущенных траекторий равно размерности фазового пространства. Чтобы это осуществить, используют в основном численный подход, основанный на выводе уравнений динамики системы в вариациях [1]. Поскольку старший показатель Ляпунова оказывает существенное влияние на эволюцию *всех* возмущенных траекторий, то на каждом этапе алгоритма кроме перенормировки векторов возмущения необходимо дополнительно выполнять их ортогонализацию. Однако такой метод требует наличия динамических уравнений системы и не подходит при обработке экспериментальных данных.

Рассмотрим процедуру численного определения спектра Ляпунова динамической системы. Пусть, для простоты изложения, фазовое пространство исследуемой динамической системы трехмерно.

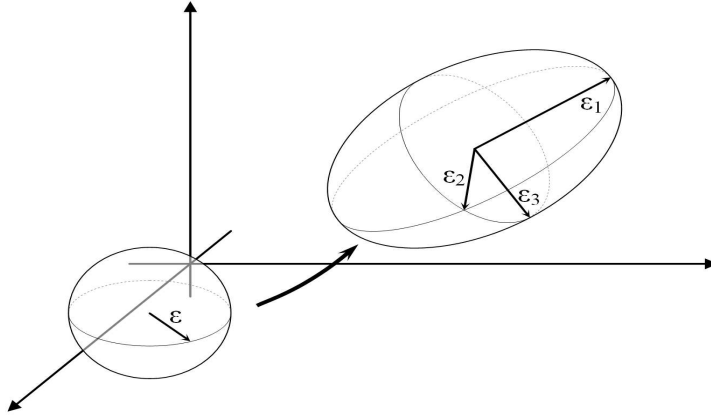


Рис. 11. К определению понятия спектра Ляпунова (случай трехмерного фазового пространства)

Пусть \mathbf{r}_0 — некоторая точка аттрактора системы. Зафиксировав некоторое небольшое, по сравнению с линейными размерами аттрактора, положительное число ε , выберем возмущенные точки \mathbf{x}_0 , \mathbf{y}_0 и \mathbf{z}_0 так, чтобы векторы возмущений $\Delta\mathbf{x}_0 = \mathbf{x}_0 - \mathbf{r}_0$, $\Delta\mathbf{y}_0 = \mathbf{y}_0 - \mathbf{r}_0$ и $\Delta\mathbf{z}_0 = \mathbf{z}_0 - \mathbf{r}_0$ имели длину ε и были взаимно ортогональны. Через некоторый небольшой промежуток времени T точки \mathbf{r}_0 , \mathbf{x}_0 , \mathbf{y}_0 и \mathbf{z}_0 эволюционируют в точки \mathbf{r}_1 , \mathbf{x}_1 , \mathbf{y}_1 и \mathbf{z}_1 , соответственно. Рассмотрим новые векторы возмущений $\Delta\mathbf{x}_1 = \mathbf{x}_1 - \mathbf{r}_1$, $\Delta\mathbf{y}_1 = \mathbf{y}_1 - \mathbf{r}_1$ и $\Delta\mathbf{z}_1 = \mathbf{z}_1 - \mathbf{r}_1$. Выполним их ортогонализацию, используя известный из линейной алгебры метод Грама-Шмидта:

$$\begin{aligned} \Delta\mathbf{x}'_1 &= \Delta\mathbf{x}_1, & \Delta\mathbf{x}''_1 &= \frac{\Delta\mathbf{x}'_1}{\|\Delta\mathbf{x}'_1\|}, \\ \Delta\mathbf{y}'_1 &= \Delta\mathbf{y}_1 - (\Delta\mathbf{y}_1, \Delta\mathbf{x}'_1)\Delta\mathbf{x}'_1, & \Delta\mathbf{y}''_1 &= \frac{\Delta\mathbf{y}'_1}{\|\Delta\mathbf{y}'_1\|}, \\ \Delta\mathbf{z}'_1 &= \Delta\mathbf{z}_1 - (\Delta\mathbf{z}_1, \Delta\mathbf{x}''_1)\Delta\mathbf{x}''_1 - (\Delta\mathbf{z}_1, \Delta\mathbf{y}''_1)\Delta\mathbf{y}''_1, & \Delta\mathbf{z}''_1 &= \frac{\Delta\mathbf{z}'_1}{\|\Delta\mathbf{z}'_1\|}. \end{aligned} \quad (29)$$

После такой ортогонализации полученные векторы возмущения будут ортонормированными, то есть будут взаимно ортогональны и иметь единичную длину. Теперь перенормируем ортогонализированные векторы возмущений так, чтобы их длины были равны ε :

$$\Delta\mathbf{x}'''_1 = \Delta\mathbf{x}''_1 \cdot \varepsilon, \quad \Delta\mathbf{y}'''_1 = \Delta\mathbf{y}''_1 \cdot \varepsilon, \quad \Delta\mathbf{z}'''_1 = \Delta\mathbf{z}''_1 \cdot \varepsilon. \quad (30)$$

Рассмотрим новый набор возмущенных точек:

$$\mathbf{x}'_1 = \mathbf{x}_1 + \Delta \mathbf{x}'''_1, \quad \mathbf{y}'_1 = \mathbf{y}_1 + \Delta \mathbf{y}'''_1, \quad \mathbf{z}'_1 = \mathbf{z}_1 + \Delta \mathbf{z}'''_1. \quad (31)$$

Далее процесс повторяется, но вместо точек $\mathbf{r}_0, \mathbf{x}_0, \mathbf{y}_0$ и \mathbf{z}_0 рассматриваются точки $\mathbf{r}_1, \mathbf{x}'_1, \mathbf{y}'_1$ и \mathbf{z}'_1 , соответственно.

Повторив описанную процедуру M раз, вычисляем суммы:

$$S_1 = \sum_{k=1}^M \ln \|\Delta \mathbf{x}'_k\|, \quad S_2 = \sum_{k=1}^M \ln \|\Delta \mathbf{y}'_k\|, \quad S_3 = \sum_{k=1}^M \ln \|\Delta \mathbf{z}'_k\|. \quad (32)$$

После этого спектр Ляпунова $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ вычисляется по формуле:

$$\lambda_i = \frac{S_i}{MT}, \quad i = 1, 2, 3. \quad (33)$$

В данном методе необходимо особое внимание уделить выбору временного промежутка T . Действительно, если взять слишком большой промежуток, то все возмущенные траектории будут отклоняться в направлении, соответствующем максимальному показателю Ляпунова и результаты вычисления будут искажены.

Описанный метод можно применять в основном только если известны уравнения эволюции системы. Но для экспериментальных данных эти уравнения, как правило, неизвестны. Далее рассмотрим методы численной оценки спектра Ляпунова, которые не требуют знания уравнений эволюции системы.

Нейросетевой метод оценки спектра Ляпунова на основе полной выборки

Основная проблема при вычислении показателей Ляпунова состоит в трудности отслеживания эволюции расстояния между точками фазовой траектории с течением времени. Поэтому разработка эффективного и робастного метода определения спектра Ляпунова является важной задачей. Для ее решения можно предложить простой метод, основанный на применении нейронной сети для вычисления расстояний между точками фазовой траектории с течением времени [29–36]. Пусть состояние динамической системы в каждый момент времени задается n -мерным вектором

$$\mathbf{X}(t) = [X_1(t), X_2(t), \dots, X_n(t)]. \quad (34)$$

Предположим, что известны временные зависимости $X_i(t)$ соответствующих переменных на определенном интервале времени. Тогда построим нейронную сеть, которая на основе предыдущего состояния динамической системы прогнозирует следующее состояние. Она будет состоять из n входных, m скрытых и n выходных нейронных элементов (рис. 12).

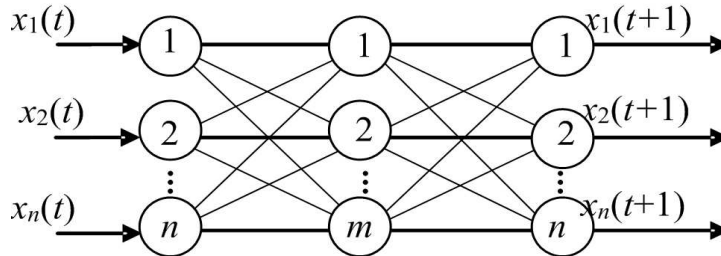


Рис. 12. Прогнозирующая нейронная сеть

Выходные значения нейронной сети определяются следующим образом:

$$X(t + 1) = F(X(t)). \quad (35)$$

После обучения такой сети можно определить как состояние динамической системы в произвольный момент времени, так и эволюцию точек фазовой траектории, используя только наблюдаемые реализации. В процессе вычислений необходимо проводить процедуру ортогонализации Грама-Шмидта. Пусть $|w_i(t)|$ — длина i -го вектора в момент времени t . Она характеризует размер вектора вдоль i -ой оси эллипсоида. Тогда i -й показатель Ляпунова можно определить следующим образом:

$$\lambda_i = \lim_{p \rightarrow \infty} \frac{1}{p} \sum_{t=1}^p \ln \frac{|w_i(t)|}{|w_i(t-1)|}. \quad (36)$$

Определяя соответствующие длины векторов $|w_i(t)|$ при помощи нейронной сети, можно вычислить спектр Ляпунова.

Рассмотрим алгоритм вычисления показателей Ляпунова. Он состоит из следующих шагов:

1. Выбираем начальную точку из области притяжения аттрактора, координаты которой равняются:

$$N(0) = [x_1(0), x_2(0), \dots, x_n(0)]. \quad (37)$$

2. Задаем достаточно малое число $\varepsilon = 10^{-8}$ и определяем координаты следующих n точек в соответствии с правилом:

$$\begin{aligned} A_1(0) &= [x_1(0) + \varepsilon, x_2(0), \dots, x_n(0)], \\ A_2(0) &= [x_1(0), x_2(0) + \varepsilon, \dots, x_n(0)], \\ &\dots\dots\dots \\ A_n(0) &= [x_1(0), x_2(0), \dots, x_n(0) + \varepsilon]. \end{aligned} \quad (38)$$

В результате выполнения этого пункта получаем следующий набор ортогональных векторов:

$$\begin{aligned} NA_1(0) &= (\varepsilon, 0, \dots, 0), \\ NA_2(0) &= (0, \varepsilon, \dots, 0), \\ &\dots\dots\dots \\ NA_n(0) &= (0, 0, \dots, \varepsilon). \end{aligned} \quad (39)$$

3. Определяем длину каждого вектора:

$$|NA_i(0)| = |W_i(0)| = \varepsilon, \quad (40)$$

где $i = \overline{1, n}$.

4. Последовательно подаем на входы нейронной сети координаты точек $N(0), A_1(0), A_2(0), \dots, A_n(0)$ в момент времени $t = 0$. В результате получаем на выходах нейронной сети координаты точек в следующий момент времени $t = t + 1$:

$$\begin{aligned} N(0) &= [x_1(1, N), x_2(1, N), \dots, x_n(1, N)], \\ A_1(0) &= [x_1(1, A_1), x_2(1, A_1), \dots, x_n(1, A_1)], \\ A_2(0) &= [x_1(1, A_2), x_2(1, A_2), \dots, x_n(1, A_2)], \\ &\dots\dots\dots \\ A_n(0) &= [x_1(1, A_n), x_2(1, A_n), \dots, x_n(1, A_n)], \end{aligned} \quad (41)$$

где $x_i(1, A_j)$ есть i -я координата точки A_j в момент времени $t = 1$.

В результате выполнения этого пункта получаем следующий набор неортогональных векторов:

$$\begin{aligned} NA_1(1) &= W_1(1) = (w_{11}, w_{21}, \dots, w_{n1}), \\ NA_2(1) &= W_2(1) = (w_{12}, w_{22}, \dots, w_{n2}), \\ &\dots\dots\dots \\ NA_n(1) &= W_n(1) = (w_{1n}, w_{2n}, \dots, w_{nn}), \end{aligned} \quad (42)$$

Здесь w_{ij} — i -я координата j -го вектора. Она определяется следующим образом:

$$w_{ij} = x_i(1, A_j) - x_i(1, N). \quad (43)$$

5. В соответствии с процедурой Грама-Шмидта производим преобразование базиса $[W_1(1), W_2(1), \dots, W_n(1)]$ в ортонормированный базис. Для этого выполняем следующие действия:

- в качестве первого вектора ортонормированного базиса выберем

$$W'_1(1) = \left[\frac{w_{11}}{|W_1(1)|}, \frac{w_{21}}{|W_1(1)|}, \dots, \frac{w_{n1}}{|W_1(1)|} \right], \quad (44)$$

где $|W_1(1)| = \sqrt{w_{11}^2 + w_{21}^2 + \dots + w_{n1}^2}$;

- остальные векторы определяются рекурсивным способом в соответствии со следующими выражениями

$$W_i(1) = W_i(1) - \sum_{j=1}^{i-1} [W_i^T(1)W'_j(1)]W'_j(1), \quad (45)$$

$$|W_i(1)| = \sqrt{w_{1i}^2 + w_{2i}^2 + \dots + w_{ni}^2}, \quad (46)$$

$$W'_i(1) = \left[\frac{w_{1i}}{|W_i(1)|}, \frac{w_{2i}}{|W_i(1)|}, \dots, \frac{w_{ni}}{|W_i(1)|} \right], \quad (47)$$

где $i = \overline{2, n}$.

6. Для каждого полученного вектора находим логарифмы отношения длин векторов в два последовательных момента времени:

$$S_i(1) = \ln \frac{|W_i(1)|}{|W_i(0)|}, \quad (48)$$

где $i = \overline{1, n}$.

7. Формируем новый набор точек, координаты которых определяются следующим образом:

$$\begin{aligned} N(1) &= [x_1(1, N), x_2(1, N), \dots, x_n(1, N)], \\ A_1(1) &= [\bar{x}_1(1, A_1), \bar{x}_2(1, A_1), \dots, \bar{x}_n(1, A_1)] \\ A_2(1) &= [\bar{x}_1(1, A_2), \bar{x}_2(1, A_2), \dots, \bar{x}_n(1, A_2)], \\ &\dots\dots\dots \\ A_n(1) &= [\bar{x}_1(1, A_n), \bar{x}_2(1, A_n), \dots, \bar{x}_n(1, A_n)], \end{aligned} \quad (49)$$

где $\bar{x}_j(1, A_j) = \varepsilon \cdot w'_{ij} + x_i(1, N)$.

8. Продолжаем процедуру, начиная с пункта 4 для $t = \overline{1, p}$, где $p \approx 1000$.
 9. Определяем спектр Ляпунова в соответствии со следующим выражением:

$$\lambda_i = \frac{1}{p} \sum_{t=1}^p S_i(t), \quad (50)$$

где $i = \overline{1, n}$.

Определяемые при этом показатели Ляпунова упорядочены по убыванию

$$\lambda_1 > \lambda_2 > \dots > \lambda_n. \quad (51)$$

На рис. 13, 14 и 15 изображены экспериментальные исследования по определению спектра Ляпунова рассмотренным выше методом для хаотических процессов Энона и Лоренца.

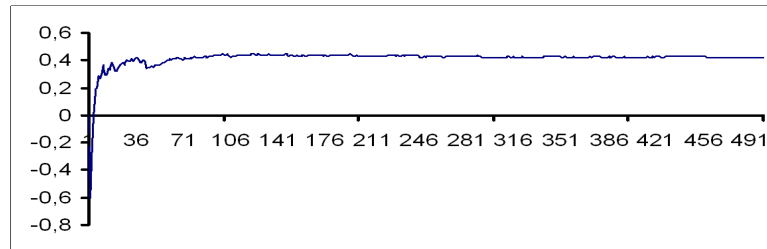


Рис. 13. Вычисление первого показателя Ляпунова для рядов Энона

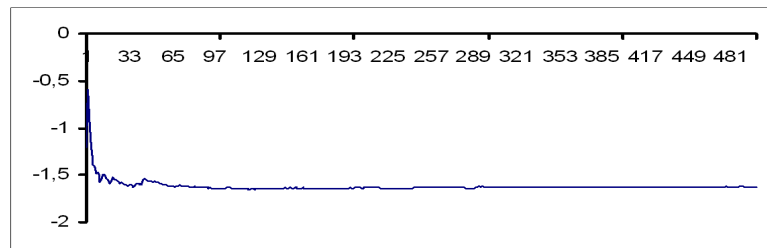


Рис. 14. Вычисление второго показателя Ляпунова для рядов Энона

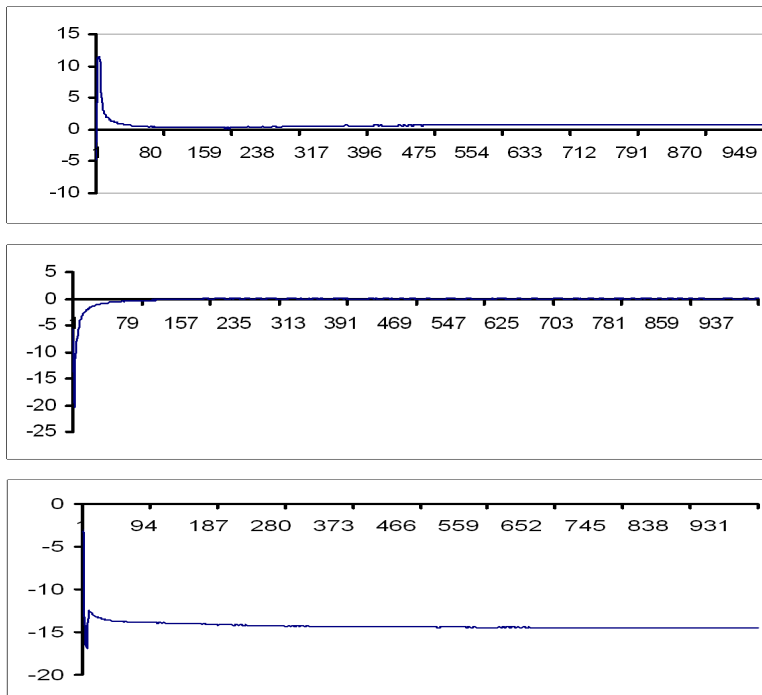


Рис. 15. Вычисление показателей Ляпунова для ряда Лоренца

Как видно из этих рисунков, для процесса Энона были получены следующие показатели Ляпунова: 0.442 и -1.625 , что хорошо согласуется с эталонными значениями 0.418 и -1.622 . Соответственно для процесса Лоренца спектр Ляпунова составил 0.777, 0.003 и -14.472 , что также достаточно точно соответствует эталонным значениям 0.906, 0, -14.472 . Таким образом, разработанный в данном разделе нейросетевой метод позволяет определить спектр Ляпунова, основываясь только на наблюдаемых реализациях.

Нейросетевой метод оценки спектра Ляпунова по выборке из одной координаты

В данном методе используется та же идея, что и в случае оценки экспоненты Ляпунова по выборке из одной координаты системы с использованием нейронной сети. Последовательность действий и рассуждений сохраняется, только здесь используется нейросетевой алгоритм, приведенный в предыдущем разделе.

Напомним последовательность действий:

1. Получить выборку измерений одной фазовой координаты динамической системы. Данная выборка должна адекватно отражать динамику процесса.
2. На основании выборки вычислить подходящую временную задержку τ (можно использовать следующие методы: *метод автокорреляционной функции*, *метод функции взаимной информации* или *метод среднего отклонения*).
3. Вычислить размерность m пространства вложения (можно использовать такие методы, как *метод корреляционной размерности*, *метод ложных ближайших соседей* или *гамма-тест*).
4. Выполнить псевдофазовую реконструкцию траектории методом *временных задержек*. При этом получим выборку m -мерных векторов, объем которой будет немного меньше объема исходной выборки.
5. Сконструировать аппроксимирующую нейронную сеть, изображенную на рис. 12.
6. Обучить нейронную сеть вычислять очередной вектор реконструированной последовательности на основании предыдущего. Обучение можно считать удачным, если ошибка вычислений каждого вектора в среднем имеет порядок 10^{-8} – 10^{-9} .
7. Применяя обученную таким образом сеть в соответствии с разделом «Нейросетевой метод оценки спектра Ляпунова на основе полной выборки», рассчитать оценку спектра Ляпунова.

Результаты экспериментов

Для апробации метода были выбраны классические хаотические системы малой размерности: система Лоренца и система Рёсслера [39, 40] Система Лоренца описывается системой трех нелинейных дифференциальных

уравнений первого порядка:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = rx - y - xz, \\ \frac{dz}{dt} = yx - bz, \end{cases} \quad (52)$$

где σ, b, r – параметры системы. При значениях параметров $\sigma = 10, b = 8/3$ и $r = 28$ динамика системы приобретает хаотический характер. Данная система хорошо изучена и все ее характеристики известны: спектр Ляпунова имеет вид $(0.906, 0, -14.572)$, фрактальная размерность аттрактора составляет 2.06.

Вторым хаотическим процессом была выбрана система Рёсслера. Она также описывается системой трех нелинейных дифференциальных уравнений первого порядка:

$$\begin{cases} \frac{dx}{dt} = -y - z, \\ \frac{dy}{dt} = x + ay, \\ \frac{dz}{dt} = b + (x - r)z, \end{cases} \quad (53)$$

где a, b, r – параметры системы. При значениях параметров $a = b = 0.2, r = 5.7$ динамика системы приобретает хаотический характер. Динамические характеристики системы Рёсслера: спектр Ляпунова $(0.07, 0, -5.39)$, фрактальная размерность аттрактора равна 2.03.

Во всех опытах выборка состояла из 400 измерений первой координаты соответствующей системы. В аппроксимирующей нейронной сети использовалось 10 нейронов. Изменяемыми параметрами в серии экспериментов были шаг дискретизации dt и временная задержка τ . Нейронная сеть в каждом эксперименте обучалась в течение 3000 эпох. В табл. 3.1 и 3.2 приведены результаты расчетов, в которых нейронная сеть обучилась достаточно хорошо. Как видно из этих таблиц, погрешность оценки вполне приемлема. Основным требованием к выборке является адекватность отражения динамики процесса: выборка должна состоять из точек, равномерно и достаточно плотно распределенных на аттракторе. Однако надо заметить, что в некоторых случаях (например, в случае системы Лоренца) метод существенно зависит от значения временной задержки, в то время как для

системы Рёсслера временная задержка может выбираться в сравнительно широком диапазоне.

Таблица 1. Результаты оценки спектра Ляпунова системы Лоренца по одной координате с использованием нейронной сети

dt	τ	Спектр Ляпунова			Ошибка	
		λ_1	λ_2	λ_3	Абсолютная	Относит.
0,04170	0,1668	0,612978	–0,2016840	–15,0033	0,559053	3,83%
0,04200	0,1680	0,725777	–0,0211582	–14,6402	0,193839	1,33%
0,04215	0,1686	0,966544	–0,3009800	–15,9458	1,407730	9,64%
0,04220	0,1688	0,965399	–0,3006240	–15,9270	1,389170	9,51%
0,08500	0,1700	1,143851	–0,2816490	–14,9843	0,553092	3,79%
0,04260	0,1704	1,021790	–0,4326160	–15,6514	1,168620	8,00%
0,04260	0,1704	0,483841	0,0528098	–13,3949	1,251610	8,57%
0,04300	0,1720	0,742471	–0,0865899	–14,2650	0,358420	2,45%
0,08600	0,1720	0,830438	–0,3357490	–13,5627	1,066370	7,30%
0,04320	0,1728	0,570654	–0,1465600	–14,9297	0,511766	3,51%
0,08700	0,1740	1,216890	–0,6435080	–14,5374	0,715508	4,90%

Прогнозирование и реконструкция хаотических процессов

Прогнозирование хаотических временных рядов

Цель предсказания временного ряда может быть описана следующим образом: для данной последовательности $x(1), x(2), \dots, x(l)$ необходимо найти ее продолжение $x(l+1), x(l+2), \dots$. Нелинейная модель для предсказания может быть представлена как $x(t) = F[x(t-1), x(t-2), \dots, x(t-k)]$, где $t = \overline{k+1, N}$, F – нелинейная функция, построенная с помощью искусственной нейронной сети, а k – размер скользящего окна, который равен числу входных нейронов сети. В качестве базовой архитектуры для прогнозирования временных процессов использовался многослойный персептрон. Как уже отмечалось, для реконструкции аттрактора с фрактальной размерностью d необходимо выбрать подходящую временную задержку и

ТАБЛИЦА 2. Результаты оценки спектра Ляпунова системы Рёсслера по одной координате с использованием нейронной сети

dt	τ	Спектр Ляпунова			Ошибка	
		λ_1	λ_2	λ_3	Абсолютная	Относит.
0,04	0,04	0,173003	-0,0821049	-5,47571	0,154879	2,87%
0,07	0,07	0,060350	-0,3888620	-5,18352	0,441825	8,19%
0,06	0,12	0,090696	0,0030709	-5,02998	0,363565	6,74%
0,06	0,12	0,106080	-0,0358488	-5,79224	0,402378	7,46%
0,06	0,12	0,077922	-0,0187908	-5,93021	0,537581	9,97%
0,07	0,14	0,129117	-0,1092460	-4,93167	0,477637	8,86%
0,08	0,16	0,106981	-0,0449128	-5,36074	0,065971	1,22%
0,08	0,16	0,085461	-0,0282390	-5,31476	0,084427	1,57%
0,04	0,16	0,119605	-0,2027930	-5,56896	0,272851	5,06%
0,06	0,18	0,141245	-0,0751598	-5,48983	0,141277	2,62%
0,08	0,48	0,078753	-0,0144016	-5,24691	0,147000	2,73%

размерность пространства вложения

$$m \geq 2[d] + 1, \quad (54)$$

где $[d]$ — целая часть фрактальной размерности. Определение параметров вложения обеспечивает максимальную предсказуемость хаотического процесса и позволяет сгенерировать обучающую выборку, а также выбрать количество входных нейронов (размер окна) прогнозирующей нейронной сети. Размер окна характеризует количество элементов ряда, одновременно подаваемых на нейронную сеть и определяется как

$$k \geq m - 1. \quad (55)$$

Тогда обучающая выборка будет иметь следующий вид:

$$\begin{aligned} Y &= x(t + i\tau), \\ X &= [x(t + (i - 1)\tau), x(t + (i - 2)\tau), \dots, x(t + (i - k)\tau)], \end{aligned} \quad (56)$$

где $i = \overline{1, n}$, Y — выходное значение нейронной сети, а X — входное значение нейронной сети для соответствующего момента времени.

После обучения нейронной сети можно прогнозировать хаотический сигнал на упреждающий промежуток времени.

Рассмотрим более детально применение многослойного персептрона для прогнозирования хаотических временных рядов. В качестве исследуемых хаотических систем будем использовать аттракторы Лоренца и Энона [1].

Последовательность значений координаты x , полученная путем численного решения системы Лоренца с использованием метода Рунге-Кутты четвертого порядка с шагом 0.01, изображена на рис. 16.

Как было упомянуто выше, для достижения максимальной предсказуемости необходимо определить параметры вложения. Согласно методу взаимной информации, для процесса Лоренца $\tau = 0.16$. Соответственно, применением метода ложных ближних соседей была получена размерность пространства вложения $m = 5$. Из этого следует, что размер окна должен быть равен $k \geq (m - 1) = 4$.

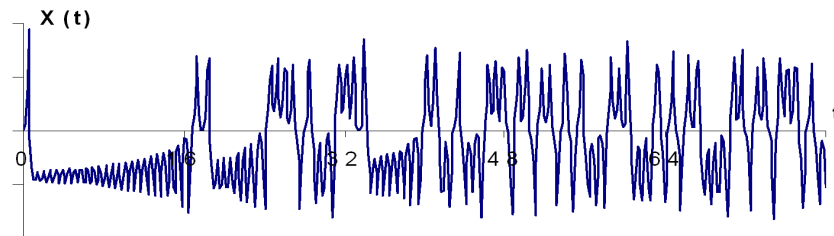


Рис. 16. Оригинальный процесс Лоренца (координата x)

Процесс Энона описывается следующими уравнениями:

$$\begin{cases} x_{n+1} = 1 - \alpha x_n^2 + y_n, \\ y_{n+1} = \beta x_n, \end{cases} \quad (57)$$

где $\alpha = 1.4$ и $\beta = 0.3$ для хаотического поведения системы.

На рис. 17 изображен временной ряд Энона (координата x). Для данного процесса были получены параметры вложения $k \geq 2$ и $\tau = 1$.

Для прогнозирования вышеуказанных хаотических рядов используем многослойный персептрон с 7 входными элементами, 5 скрытыми нейронами с сигмоидной функцией активации, одним линейным выходным нейроном. В качестве метода обучения использовался алгоритм обратного распространения ошибки с адаптивным шагом. Размерность тренировоч-

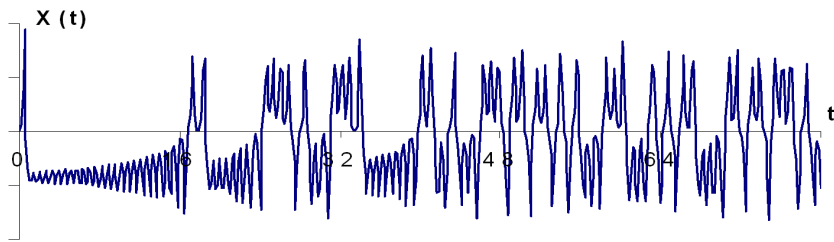


Рис. 17. Последовательность Энона (первые 200 элементов ряда)

ного множества составляет 930 образов для данных Энона и 800 образов для данных Лоренца.

Рис. 18 и 19 показывают результаты предсказания на 30 шагов для временных рядов Энона и Лоренца, соответственно. Как видно из этих рисунков, предсказание на уровне индивидуальных точек не является точным. В этом проявляются основные свойства хаотических систем.

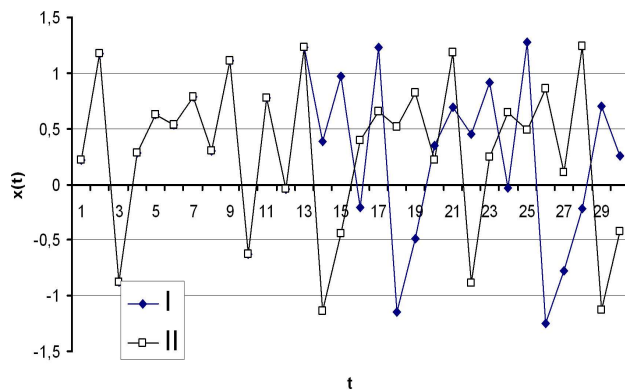


Рис. 18. Прогноз процесса Энона для 30 итераций прогноза: I — прогноз, II — оригинальная последовательность

Рассмотрим определение периода прогнозирования для хаотических процессов, который назовем горизонтом прогнозирования. Горизонт про-

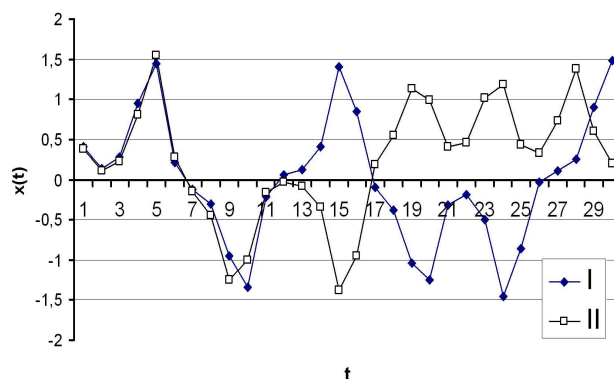


Рис. 19. Прогноз процесса Лоренца для 30 итераций прогноза: I — прогноз, II — оригинальная последовательность

гнозирования характеризует интервал времени T , в течение которого возможно совершить точный прогноз. Как было отмечено раньше, хаотические данные непредсказуемы на большой промежуток времени, так как ошибка измерения в начальных условиях растет экспоненциально по времени. Такая высокая чувствительность к начальным условиям определяется положительным показателем Ляпунова. Следовательно, положительное значение наибольшего показателя Ляпунова определяет верхнюю границу периода предсказуемости. Как известно, сумма положительных показателей Ляпунова равна энтропии Колмогорова [1,4]. Тогда, в соответствии с теорией хаоса [4], горизонт прогнозирования может быть представлен следующим образом:

$$T \approx \frac{1}{K} \cdot \ln \left(\frac{1}{d_0} \right), \quad (58)$$

где $K = \sum_i \lambda_i$ — энтропия Колмогорова и $\lambda_i > 0$, d_0 — начальная ошибка прогноза.

В соответствии с уравнением (58) точный прогноз возможен только на период T . Таким образом, после обучения нейронной сети можно найти период корректного прогноза для хаотического ряда.

Рассмотрим нейросетевой подход для увеличения горизонта прогнозирования. Как было отмечено выше, длина прогноза для хаотического по-

ведения ограничена в соответствии с уравнением (58). Одним из методов увеличения периода прогнозируемости является подход с переобучением нейронной сети. Рассмотрим данный подход более детально.

Предположим, что нейронная сеть обучена с использованием следующего набора данных:

$$X = \{x(1), x(2), \dots, x(N)\}. \quad (59)$$

В соответствии с выражением (58) можно сравнительно точно предсказать хаотический временной ряд на T точек вперед. В результате этого мы определяем следующие предсказанные точки:

$$x(N + 1), x(N + 2), \dots, x(N + T). \quad (60)$$

На следующем этапе организуем новый обучающий набор данных в виде:

$$X' = \{x(1), x(2), \dots, x(N + T)\}. \quad (61)$$

Переобучая нейронную сеть для нового набора данных, можно увеличить период корректного прогноза.

Для тестирования предложенной методики были проведены эксперименты с использованием процессов Энона и Лоренца, свойства которых детально описаны в начальной части данного раздела. Таблицы 3 и 4 иллюстрируют сравнительные результаты итерационного подхода и подхода с переобучением при прогнозировании данных Энона. В данных таблицах MSE1 и MSE2 — среднеквадратические ошибки для спрогнозированных точек $x(N + 1), x(N + 2), x(N + 3), x(N + 4)$ и $x(N + 5), x(N + 6), x(N + 7), x(N + 8)$, соответственно, NIT — количество итераций обучения, STS — размер множества обучения. Таблицы 5 и 6 показывают аналогичные результаты для данных Лоренца.

Как видно, подход с переобучением позволяет получить лучшие результаты прогноза в обоих случаях по сравнению с итерационным и тем самым увеличить длину прогноза.

Реконструкция аттракторов

Прогнозирующие свойства нейронных сетей можно использовать для построения странных аттракторов динамических систем. Странные аттракторы имеют фрактальную размерность и характеризуют хаотические процессы.

Таблица 3. Результаты обучения сети для данных Энона

Метод	NIT	STS	MSE	T	MSE1	MSE2
Итерационный	308	950	$3 \cdot 10^{-4}$	4	0.00022275	0.031198
С переобучением	276	954	$3 \cdot 10^{-4}$	4	0.00003325	0.0080427

Таблица 4. Результаты прогнозирования для данных Энона

Метод	Прогноз	Оригинальные данные	Абсолютная ошибка
Итерационный	0.365621	0.363170	0.002451
	0.992627	1.002511	0.009884
	-0.274204	-0.298088	0.023884
	1.191078	1.176354	0.014724
	-1.101723	-1.026758	0.074965
	-0.363043	-0.123019	0.240024
	0.512435	0.670785	0.158350
С переобучением	0.524174	0.333160	0.191014
	0.364677	0.363170	0.001507
	1.001295	1.002511	0.001216
	-0.288775	-0.298088	0.009313
	1.182933	1.176354	0.006579
	-1.046040	-1.026758	0.019282
	-0.247162	-0.123019	0.124143
С переобучением	0.592083	0.670785	0.078702
	0.434126	0.333160	0.100966

Пусть дана временная последовательность только одной составляющей хаотического процесса. Тогда для построения аттракторов можно использовать, как уже отмечалось, метод псевдофазового пространства. Он базируется на теореме Такенса [13], который показал, что можно восстановить некоторые свойства аттрактора в фазовом пространстве, используя временную последовательность только одной составляющей. Метод псевдофазового пространства заключается в том, что можно реконструировать топологию аттрактора, исходя из наблюдения сигнала $X(t)$ и выбирая в качестве системы координат $X(t)$, $X(t + \tau)$ и т. д.

Отображая спрогнозированные значения $X(t)$ на псевдофазовую плоскость, можно построить соответствующий аттрактор динамической системы. Таким образом, общая методика построения аттракторов при помощи

Таблица 5. Результаты обучения сети для данных Лоренца

Метод	NIT	STS	MSE	T	MSE1	MSE2
Итерационный	1000	800	0.00135	5	0.0053618	0.1628954
С переобучением	578	805	0.0014	5	0.0011142	0.0698684

нейронных сетей состоит из следующих пунктов:

1. Определение количества входных нейронов сети $k \geq (m - 1)$.
2. Подготовка обучающей выборки в соответствии с временной задержкой τ и размером окна k .
3. Обучение нейронной сети.
4. Осуществление многошагового прогноза.
5. Отображение спрогнозированных точек на псевдофазовую плоскость.

Использование нейронных сетей позволяет построить аттрактор динамической системы, используя малый объем временной последовательности. Рассмотрим это на примерах. На рис. 20 и 21 изображены оригинальные аттракторы Лоренца и Энона, соответственно.

Произведем обучение нейронной сети, используя обучающую выборку размером в 100 элементов для данных Энона и 200 элементов для данных Лоренца. После 3000 итераций обучения суммарная ошибка составила 0.00033 и 0.0008 для данных Энона и Лоренца, соответственно.

Основываясь на итерационном подходе, был выполнен прогноз данных Лоренца и Энона на 1500 шагов. Аттракторы Энона и Лоренца, построенные на спрогнозированных данных, представлены на рис. 22 и 23. Очевидно, что многослойная нейронная сеть обобщает свойства хаотического поведения системы и, следовательно, может быть применена для псевдофазовой реконструкции. Таким образом, нейронная сеть способна предсказывать поведение сложной системы.

В данном разделе рассмотрен нейросетевой подход к реконструкции фазового пространства. Он позволяет строить аттракторы хаотических систем, используя малый объем экспериментальных данных.

Общая методика обработки хаотических процессов

Рассмотрим общую методику обработки хаотических процессов, если известны только временные последовательности, характеризующие динами-

Таблица 6. Результаты прогнозирования для данных Лоренца

Метод	Прогноз	Оригинальные данные	Абсолютная ошибка
Итерационный	–0.155480	–0.163600	0.008120
	–0.556713	–0.617800	0.061087
	–1.573766	–1.633100	0.059334
	–0.536221	–0.439700	0.096521
	0.085535	0.186400	0.100865
	0.237657	0.520500	0.282843
	0.719185	1.254000	0.534815
	1.509935	0.938200	0.571735
	0.461715	0.245600	0.216115
С переобучением	–0.042810	0.230800	0.273610
	–0.169124	–0.163600	0.005524
	–0.613167	–0.617800	0.004633
	–1.598149	–1.633100	0.034951
	–0.430258	–0.439700	0.009442
	0.121533	0.186400	0.064867
	0.317614	0.520500	0.202886
	0.940051	1.254000	0.313949
	1.336355	0.938200	0.398155
0.301510	0.245600	0.055910	
0.011798	0.230800	0.219002	

ческую систему. Цель данной методики состоит в идентификации хаотического поведения динамической системы, прогнозирования ее временных характеристик на упреждающий промежуток времени и реконструкция фазовой траектории системы. Пусть известны характеристики динамической системы в виде n или одной временных последовательностей:

$$\mathbf{X}(t) = [X_1(t), X_2(t), \dots, X_n(t)], \quad (62)$$

или

$$X(t) = X_i(t), \quad t = \overline{1, p}. \quad (63)$$

Тогда методика обработки хаотических процессов будет состоять из следующих шагов:

1. Выбираем любую временную последовательность $X_i(t)$, $t = \overline{1, p}$.
2. Определяем временную задержку τ и располагаем отсчеты временной последовательности через промежуток, равный τ .
3. Определяем размерность пространства вложения m .

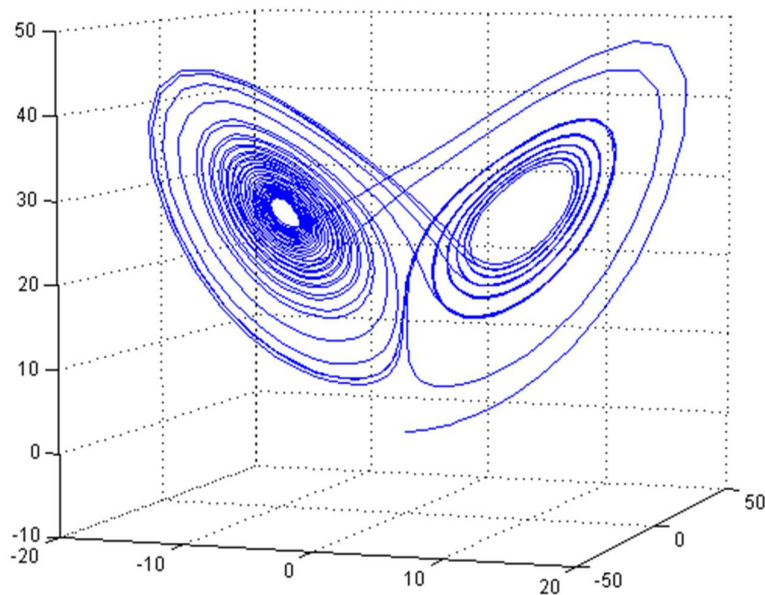


Рис. 20. Трехмерный аттрактор Лоренца

4. Выбираем архитектуру многослойного персептрона, которая состоит из $k \geq (m - 1)$ входных, l скрытых и одного выходного нейронных элементов.
5. Формируем обучающую выборку:

$$\mathbf{X}(t) = [X_i(t), X_i(t + 1), \dots, X_i(t + k - 1)], Y(t) = X_i(t + k), \quad (64)$$
 где $\mathbf{X}(t)$ – вектор входных значений нейронной сети, а $Y(t)$ – соответствующее выходное значение $t = \overline{1, p}$.
6. Обучаем нейронную сеть, используя алгоритм обратного распространения ошибки.
7. Определяем наибольший показатель Ляпунова с использованием нейронной сети и идентифицируем наличие хаоса в динамической системе.

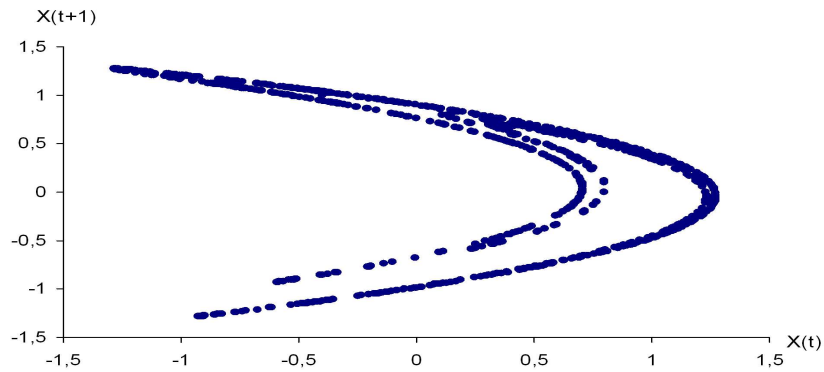


Рис. 21. Оригинальный аттрактор Энона, построенный для 1500 элементов

8. Если известны n временных характеристик динамической системы, где $n = \frac{(m-1)}{2} + 1$, то определяем спектр Ляпунова в соответствии с разделом «Нейросетевой метод оценки спектра Ляпунова на основе полной выборки», в противном случае определяем спектр Ляпунова в соответствии с разделом «Нейросетевой метод оценки спектра Ляпунова по выборке из одной координаты»
9. Прогнозируем временную последовательность $X_i(t)$ на упреждающий промежуток времени.
10. Осуществляем псевдофазовую реконструкцию.

Приведенная выше методика предполагает применение нейронных сетей для обработки хаотических процессов, что позволяет упростить вычислительные процедуры и использовать малый объем исходной временной последовательности.

Заключение

В данной лекции рассмотрены методы обработки хаотических процессов по наблюдаемым реализациям. Проведен анализ как классических, так и нейросетевых методов обработки хаотических процессов. Представлены следующие нейросетевые подходы для обработки хаотических процессов:

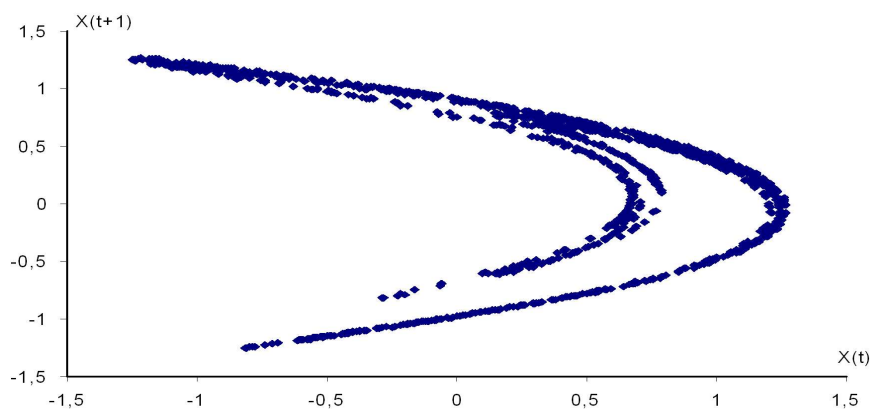


Рис. 22. Аттрактор Энона, построенный для 1500 итераций прогноза

1. Нейросетевой метод для вычисления наибольшего показателя Ляпунова, который базируется на применении многослойного персептрона для вычисления расхождения двух близлежащих траекторий. Предложенный метод позволяет вычислять искомый показатель при малом объеме исходной временной последовательности и обладает меньшей вычислительной сложностью по сравнению с традиционными методами.
2. Нейросетевой метод для определения спектра Ляпунова как на основе всех фазовых координат системы, так и на основе одной фазовой координаты системы. Данный метод основывается на применении многослойного персептрона, который на основе предыдущего состояния динамической системы прогнозирует следующее состояние. Достоинством предложенного метода являются возможность определения показателей Ляпунова на основе наблюдаемых реализаций хаотического процесса.
3. Исследовано применение нейронных сетей для построения странных аттракторов на основе прогнозируемых значений хаотического процесса. В качестве исходных данных использовались последовательности Энона и Лоренца. Показано, что при помощи нейронной сети можно осуществить реконструкцию динамики системы, используя малый объем исходных данных.

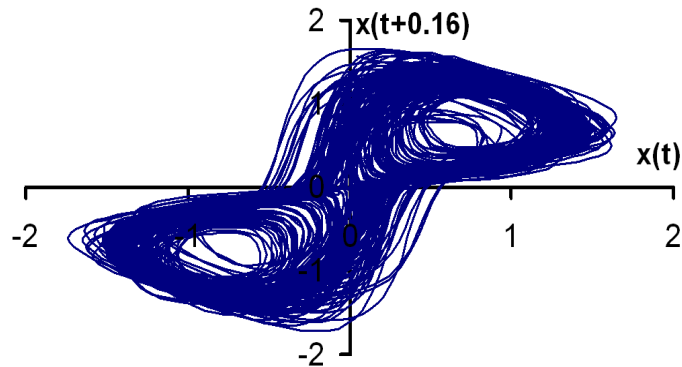


Рис. 23. Аттрактор Лоренца, построенный для 1500 итераций прогноза

4. Метод переобучения нейронной сети для увеличения горизонта прогнозирования хаотических процессов. Он заключается в том, что после обучения нейронной сети на исходных данных осуществляется прогнозирование на T шагов вперед в соответствии с уравнением (58) и формируется новая обучающая выборка, включающая спрогнозированные значения.
5. Нейросетевая методика обработки хаотических процессов, которая позволяет идентифицировать хаотическое поведение динамической системы, прогнозировать ее временные характеристики и реконструировать аттрактор. В отличие от других известных методик, она позволяет использовать малый объем исходных данных хаотического процесса.

Литература

1. Кузнецов С. Динамический хаос. – М.: Изд-во физико-математической литературы, 2001. – 236 с.
2. Пригожин И. Порядок из хаоса: Пер. с англ. – М.: Прогресс, 1986. – 432 с.
3. Хакен Г. Синергетика: Пер. с англ. – М.: Мир, 1980. – 370 с.
4. Шустер Г. Детерминированный хаос. – М.: Мир, 1980. – 317 с.

5. Prigogine I., Antoniou I. Message from honorary chairman // *Proc. Second International Conference on Neural Networks and Artificial Intelligence (ICNNAI'2001)*, Minsk, October 2001. – Minsk: BSU, 2001. – p. 8.
6. Cascadi M. Nonlinear prediction of chaotic time series // *Physica D.* – 1989. – No. 35. – pp. 335–336.
7. Kantz H., Schreiber T. Nonlinear time series analysis. – Cambridge: Cambridge University Press, 1997. – 75 pp.
8. Juda K., Small M. Towards long-term prediction // *Physica D.* – 2000. – No. 136. – pp. 31–34.
9. Eckmann J., Kamphorst S., Ruelle D., Gillerto D. Lyapunov exponents from a time series // *Phys. Review.* – 1986. – A. 34. – pp. 4971–4979.
10. Eckmann J., Ruelle D. Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems // *Physica D.* – 1992. – No. 56. – pp. 185–187.
11. Grassberger P., Procaccia I. Measuring the strangeness of strange attractors // *Physica D.* – **9**, 1983.
12. Packard N.H., Crutchfield J.P., Farmer J.D., Shaw R.S. Geometry from a time series // *Physical Review Letters.* – **45**, 1980, pp. 712–716.
13. Takens F. Detecting strange attractors in turbulence // *Lecture Notes in Mathematics*, Vol. 898, Springer-Verlag, Berlin, 1980, pp. 366–381; and in *Dynamical System in Turbulence*, Warlock, 1980, Eds. D. Rand and L. S. Young.
14. Albano A.M., Muench J., Schwartz C., Mees A.I., Rapp P.E. Syngular-value decomposition and the Grassberger-Procaccia algorithm // *Physical Review A.* – **38**, 1988, pp. 3017–3026.
15. Casdagli M., Eubank S., Farmer J.D., Gibson J. State space reconstruction in present of noise // *Physica D.* – **51**, 1992, pp. 52–98.
16. Zeng X., Eykholt R., Pielke R.A. Estimating the Lyapunov-Exponent spectrum from short time series of low precision // *Physical Review Letter.* **66**, 1991, pp. 3229–3232.
17. Holzfuss J., Mayer-Kress G. An approach to error estimation in the applications of dimensional algorithms // In: *Dimensions and Entropies in Chaotic Systems*, Ed. G. Mayer-Kress, Springer-Verlag, New York, 1986, pp. 114–122.
18. Rosenstein M. T., Colins J. J., De Luca C. J. Reconstruction expansion as a geometry-based framework for choosing proper delay time // *Physica D.* – **73**, 1994, pp. 82–98.
19. Fraser A. M., Swinney H. L. Independent coordinates for strange attractor from mutual information // *Physical Review A.* – **33**, 1986, pp. 1134–1140.

20. Abarbanel H. D. I., Brown R., Sidorovich J., Tsimring L. The analysis of observed chaotic data in physical systems // *Reviews of Modern Physics*, Vol. 65, No. 4, 1993, pp. 1331–1392.
21. Castro R., Sauer T. Correlation dimension of attractor through interspike intervals // *Physical Review E*. – 55, 1997.
22. Kennel M. B., Brown R., Abarbanel H. D. I. Determining embedding dimension for phase-space reconstruction using a geometrical construction // *Physical Review A*. – 45, 1992, pp. 3403–3411.
23. Kugiumtzis D. State space reconstruction parameters in the analysis of chaotic time series – the role of the time window length, 1996.
24. Stefansson N., Koncar N., Jones A. J. A note on the Gamma test // *Neural Computing and Applications*. – 5, 1997, pp. 387–393.
25. Benettin G., Galgani L., Giorgilli A., Strelcyn J.-M. Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems: A method for computing all of them. – Pt. I: Theory. Pt. II: Numerical applications, *Meccanica*, Vol. 15, 1980, pp. 9–30.
26. Wolf A., Swift J., Swinney H., Vastano J. Determining Lyapunov exponents from a time series // *Physica D*. – 1985. – No. 16. – pp. 285–301.
27. Golovko V., Savitsky Y., Maniakov N. Modeling nonlinear dynamics using multilayer neural networks // *Proc. of the International Workshop on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2001)*, July 1 – 4, 2001, Foros, Ukraine. – Ternopil: Lileya, 2001. – pp. 197– 202.
28. Golovko V., Savitsky Y., Maniakov N., Rubanov V. Some Aspects of Chaotic Time Series Analysis // *Proc. of the 2nd International Conference on Neural Networks and Artificial Intelligence*, October 2–5, 2001, Minsk, Belarus, pp. 66–69.
29. Головки В. А., Савицкий Ю. В. Нейросетевые методы определения спектра Ляпунова // *Международный журнал «Компьютинг»*. – 2002. – № 1. – с. 80–86.
30. Головки В. А., Чумерин Н. Ю., Савицкий Ю. В. Нейросетевой метод оценки спектра Ляпунова по наблюдаемым реализациям // *Вестник Брестского государственного технического университета*. – 2002. – №4. – с. 66–70.
31. Golovko V. From neural networks to intelligent systems: Selected aspects of training, application and evolution // Chapter of NATO book “*Limitations and Future Trends in Neural Computation*”. – Amsterdam: IOS Press, 2003, pp. 219–243.

32. *Golovko V., Savitsky Y., Maniakov N.* Neural networks for signal processing in measurement analysis and industrial applications: The case of chaotic signal processing // Chapter of NATO book “*Neural networks for instrumentation, measurement and related industrial applications*”. – Amsterdam: IOS Press, 2003, pp. 119–143.
33. *Golovko V.* Estimation of Lyapunov spectrum from one-dimensional observations using neural networks // *Proc. of the Second IEEE Int. Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2003)*, Lviv, September 8–10, 2003. – Piscataway: IEEE Service Center, 2003. – pp. 95–98.
34. *Golovko V., Savitsky Y.* Computing of Lyapunov exponents using neural networks // *Proc. of the Third International Conference on Neural Networks and Artificial Intelligence (ICNNAI'2003)*, Minsk, November 12–14, 2003. – Minsk: BSUIR.
35. Головки В., Чумерин Ю. Нейросетевые методы определения спектра Ляпунова хаотических процессов // «*Нейрокомпьютеры: Разработка и применение*», 2004. – № 1.
36. *Golovko V., Savitsky Y.* Computing of Lyapunov exponents using neural networks // *Computing*, 2004. – Vol. 3. – No. 1. – pp. 93–98.

Владимир Адамович ГОЛОВКО, доктор технических наук, профессор, заведующий кафедрой «Интеллектуальные информационные технологии» и научно-исследовательской лабораторией «Искусственные нейронные сети» Брестского государственного технического университета. Область научных интересов — искусственный интеллект и нейронные сети, хаотические процессы, обнаружение и распознавание атак на компьютерные сети, автономное управление мобильными роботами, прогнозирование. Автор трех монографий, трех глав в книгах и более 150 научных публикаций.

С. А. ТЕРЕХОВ
ООО «Нейрок Техсофт»,
г. Троицк, Московская обл.
E-mail: alife@narod.ru

**АДАПТИВНЫЕ НЕЙРОСЕТЕВЫЕ МЕТОДЫ В МНОГОШАГОВЫХ
ИГРАХ С НЕПОЛНОЙ ИНФОРМАЦИЕЙ**

Аннотация

В лекции рассматриваются понятия и постановки задач математической теории игр. Особое внимание уделяется информационным и вычислительным аспектам игрового моделирования. Для важного случая игр нескольких игроков с неполной информацией предложена вероятностная формализация алгоритма обучения игрока оптимальной смешанной стратегии принятия решений. Существенную часть лекции составляет подробное описание практического примера теоретико-игрового моделирования в оптимизации совместной бизнес-деятельности группы компаний, предоставляющих консалтинговые услуги. Модель основывается на нейросетевой аппроксимации функции цены состояния игроков. При моделировании используется нейронная сеть специального вида (локальные нормализованные сплайны), алгоритмы и иллюстрации характера обучения которой описаны в Приложении.

S. A. TEREKHOFF
Neurok Techsoft, Ltd,
Troitsk, the Moscow Region
E-mail: alife@narod.ru

**ADAPTIVE NEURAL NETWORK TECHNIQUES IN GAMES WITH
INCOMPLETE INFORMATION**

Abstract

This Lecture introduces basic topics of contemporary game theory. Simulation and informational aspects of games are emphasized. Key concept of games with incomplete information is studied from point of view of probabilistic formalism of Markov decision process. Player payoff probabilities are approximated by neural network model with specific architecture (Connectionist Normalized Local Splines). The model of strategic cooperation and competition in the internal market of holding company is served as a practical application example.

Введение: От задач оптимизации и оптимального управления к теории игр

Разобравшись в деле, убавь то, что должно быть убавлено.

*И Цзин,
Гексаграмма Сунь*

Ключевым аспектом организационной деятельности является принятие решений. В условиях, когда характер деятельности допускает полное подробное описание и всецело зависит только от ваших решений, эти решения, скорее всего, будут относиться к области *оптимизации*. Оптимизация состоит в проведении непосредственных изменений параметров деловой системы с целью максимизации выбранной функции ценности. Примером оптимизационного решения может служить пространственное размещение продукции в складском помещении, уменьшающее затраты на ее отгрузку потребителю [1].

В более сложных случаях, когда непосредственное произвольное изменение желаемых параметров затруднено, лицо, принимающее решения, имеет дело с задачей *управления*. Управляющие решения (разовые или последовательности решений) характеризуются внесением изменений в некоторые переменные или параметры системы, которые лишь опосредованно отражаются в движении системы по благоприятной траектории или к желаемому состоянию. Широкий класс задач управления основывается на измерении отклика системы на управляющие решения. Возможный пример — система управления качеством бизнес-процессов «6 sigma» [2]. Одним из центральных принципов этого подхода является проведение измерений качества и использование принципа обратной связи.

Новым масштаб сложности возникает в ситуациях, когда отклик бизнес-системы зависит одновременно от решений нескольких лиц, занятых достижением индивидуальных целей. Даже в случаях, когда каждый из участников, принимающих решения, обладает полной информацией о совместно управляемой системе, ее отклик и степень достижения целей оказываются весьма богатыми по своему разнообразию. Индивидуальные наилучшие решения могут не существовать (в математическом смысле), либо быть неустойчивыми — некоторые из участников предпочтут изменить свои решения в следующие моменты времени.

Рациональные способы принятия решений несколькими лицами являются предметом *теории игр* — математической дисциплины, замыкающей цепочку, начатую оптимизацией и теорией управления.

В соответствии с этой логической последовательностью усложнения проблемы принятия решения выстроена и серия вводных лекций, прочитанных автором на Школе по нейроинформатике в МИФИ в 2001–2004 годах. В первой из лекций рассматривалась проблема описания неопределенностей в оптимизируемой системе с позиции аппроксимации плотности вероятности ее состояний. Во второй лекции, посвященной байесовым сетям, обсуждались практические вопросы принятия решений в условиях неопределенности и моделирование вероятностей достигаемых вследствие решений состояний системы. Третья лекция обобщает механизмы принятия решений на случай многошаговых задач управления, с поиском нестационарных управляющих траекторий методами нейро-динамического программирования (обучения с подкреплением).

Новая, четвертая, лекция рассматривает математические имитационные модели принятия рациональных решений в играх, в которых игроки не имеют полной информации о системе (и о решениях других игроков). При этом основное внимание будет уделено повторяющимся играм, когда управляющие решения применяются игроками многократно, а целью служит нахождение динамических стратегий принятия решений, обеспечивающих для каждого игрока наилучший совокупный исход множества игр.

Базовым исследовательским инструментом, используемым автором во всех лекциях¹, являются искусственные нейронные сети, аппроксимирующие искомые в соответствующих задачах функции (плотность вероятности, частные условные вероятности в байесовых сетях, функции цены решения в нейродинамическом программировании). Поэтому параллельно основному содержанию, в лекциях обсуждаются также вопросы эффективного синтеза и обучения нейросетевых моделей.

В этой лекции нейронная сеть используется для моделирования распределения вероятности суммарного дисконтированного выигрыша игрока. Примененная архитектура — локально-связанные нейросетевые сплайны (Connectionist Local-Spline Neural Network — CLSNN) была предложена в середине 80-х годов в Центре исследований нелинейности Национальной лаборатории в Лос-Аламосе, США. Важной особенностью этой нейросети

¹Равно, как и в основной производственной деятельности.

является удачное сочетание локальной и глобальной аппроксимации благодаря специальной нормировке радиальных базисных функций. Подробно эта нейросеть описана в Приложении.

Теория игр, как научное направление, относительно молода (в сравнении с теорией оптимизации по Лагранжу, вариационным исчислением и оптимальным управлением) — фундаментальные положения ее были сформулированы в середине XX века Дж. фон Нейманом [3]. Примерно в это же время было установлено понятие равновесия стратегий [4] по Нэшу. В дальнейшем были предложены различные подходы к достижению равновесия [5]. Цикл исследований по применениям теории игр в экономических задачах был отмечен Нобелевской премией по экономике (Дж. Нэш, Дж. Харшани, Р. Зельтен, 1994). Нужно сказать, что теория игр не занимала в нашей стране такого ведущего положения как в США (в теории управления, по всей видимости, наблюдалась обратная ситуация), что во многом было вызвано особым отношением к экономическим вопросам, которые составляют основное прикладное содержание теории игр. В последнее время число публикаций и книг по теории игр в России значительно возросло [9–10 и др.], многие монографии по оптимизации и управлению включают теоретико-игровые разделы.

Приложения теории игр не ограничиваются экономическими проблемами. К ним относятся такие области, как коллективное принятие решений советом экспертов, эксплуатация совместной собственности, судебная деятельность, международные переговоры, военно-стратегическое планирование, политические и маркетинговые технологии.

Данная лекция, вводная по своей сути, призвана расширить круг интересующихся теоретико-игровыми приложениями в контексте практического использования методов обучения машин и нейроинформатики для моделирования практических игровых ситуаций в деловой и организационной деятельности.

В первом разделе будут в простой форме обсуждены основные понятия теории игр, приведены иллюстративные примеры игр. Второй раздел будет посвящен многошаговым играм, в которых решения принимаются в динамике и с учетом предыдущей истории игр. Многошаговый характер игры позволяет обобщить методы обучения с подкреплением, развитые ранее для случая одного игрока («игры с природой»). В случае, когда игроки обладают полной информацией, будет предложен вычислительный подход к поиску равновесия на основе нейродинамического программирования.

В третьем, наиболее интересном с прикладной точки зрения, разделе бу-

дет описан максимально близкий к реальности случай байесовых игр (игр с неполной информацией). Изложение будет построено вокруг имитационной модели деятельности холдинга компаний, оказывающих консалтинговые услуги. Будет показано, как можно обеспечить эффективное управление группой компаний на основе распределения выполняемых бизнес-проектов путем проведения внутренних аукционов.

Специфика теоретико-игровых постановок информационных задач

Прежде чем перейти к изложению основных понятий, с которыми оперирует теория игр, рассмотрим пример «простой» игры, в решении которой содержатся многие ключевые особенности теоретико-игровых задач.

Игра 1 (Дележ). Двое имеют возможность разделить сумму в 100 долларов. Каждый игрок, в тайне от другого игрока, сообщает судьбе (или помещает в запечатанный конверт) количество денег, которое он желает получить. Конверты вскрываются одновременно и если сумма двух значений не превышает 100, то каждый получает столько, сколько он запросил. Однако, если сумма оказывается больше 100, никто не получает ничего. Как следует² поступить игрокам?

Допустим, что игроки сделали заявки x и y . Каждому из них сообщен результат его игры (запрошенная сумма или ноль), но не ставка соперника. Следует ли игрокам изменить свои решения, если им представится возможность сыграть еще раз? Как изменится подход³ к решению, если обе ставки становятся известными обоим игрокам?

При размышлении, ответы на эти вопросы становятся все менее очевидными. В целом понятно, что если сумма ставок оказалась меньше 100, то свою ставку при следующем ходе можно увеличить каждому игроку (и уменьшить, если они не получили ничего). После нескольких ходов игры

²Предварительные переговоры между игроками не допускаются. Но даже наличие (словесных, без механизма обеспечения условий договоренности) переговоров между игроками мало что меняет — игроки могут проигнорировать итоги обсуждения.

³Игры такого типа весьма близки к проблемам использования совместной собственности. Два фермерских хозяйства располагают общим пастбищем. Каждая ферма решает, какую численность стада целесообразно содержать. При превышении некоторого предела по общему поголовью, пастбище теряет возможность восстанавливаться, и оба хозяйства терпят убытки. В пределе полного эгоизма ни одно из хозяйств не желает отступать от достигнутого исторически (одного из возможных) равновесия.

сумма ставок, вероятно, приблизится к 100. Однако дальнейшее движение не выглядит рациональным ни для одного игрока — и уменьшение, и увеличение своей ставки одним из игроков приведут к потерям, каково бы ни было достигнутое ранее распределение ставок.

Такие конфигурации решений, при которых каждый игрок не может единолично улучшить свою премию, называются состояниями равновесия (по Нэшу). Приведенная Игра 1 имеет бесконечное число состояний равновесия вида $[x, 100 - x]$.

Понятие равновесия является ключевым в теории игр — собственно, результатом теории является обнаружение и классификация состояний равновесия, равновесных стратегий (о стратегиях — см. ниже) игроков, а также получаемых ими выигрышей (т. е. вычисление цены игры).

Другим существенным моментом является само понятие рационального поведения. Формализация этого понятия проведена Дж. фон Нейманом [3] в теории рационального выбора, основанной на аксиоматическом подходе. Она состоит в том, что *рациональный* игрок применяет наилучшее для себя решение A из набора возможных решений, руководствуясь индивидуальной функцией полезности (utility function, $U(A)$). Значения функции полезности являются абстрактными и имеют только относительный смысл при сравнении двух возможных решений. Так, для двух возможных решений A и B , ситуации $\{U(A) = 2, U(B) = 1\}$ и $\{U(A) = 100, U(B) = 1\}$ рассматриваются как идентичные, т. е. предпочтение безусловно отдается решению A (при этом во втором случае это предпочтение не является ни в каком смысле «более сильным»).

Стратегические игры в матричной форме

Если игра проводится конечное и *известное заранее* число раз, то каждый игрок может предложить набор *ходов* (управляющих решений) для каждой из возможных промежуточных ситуаций в игре. Перечень решений для каждой из ситуаций игры называют *стратегией* игрока. Такая совокупность ходов, понимаемая как один макроход, по индукции, может быть выбрана до начала игры и таким образом игра фактически является одношаговой. Игроки публикуют свои стратегии и вычисляются итоговые выигрыши и цена всей игры.

Игры этого типа называются стратегическими. Выигрыш каждого игрока зависит от стратегий *всех* игроков. При известном (и конечном) наборе стратегий он может быть явно вычислен. Таким образом, все выигрыши

данного игрока для всех комбинаций стратегий формируют матрицу (с числом измерений, равным числу игроков). Индексами в каждом измерении служат номера стратегий соответствующих игроков, а матричные элементы равны выигрышам игрока. В общем случае каждый игрок имеет дело со своей матрицей выигрышей.

Для простейшего случая двух игроков матрицы выигрышей представляют собой обычные прямоугольные таблицы чисел. В стратегической игре один игрок выбирает номер строки в матрице, второй игрок — номер столбца. После этого игра считается состоявшейся, выигрыш дается соответствующим матричным элементом.

Таким образом, исход игры с матрицей выигрышей M есть, суть, произведение:

$$\vec{p}_R \cdot M \cdot \vec{p}'_C, \quad (1)$$

где, \vec{p}_R — вектор решения игрока, выбирающего строку матрицы, все компоненты вектора кроме одной (единичной) равны нулю, \vec{p}'_C — аналогичный вектор решения игрока, выбирающего столбец⁴.

Равновесие в играх с полной информацией

Действующий рационально игрок R должен стремиться к максимизации своего выигрыша (1) при любых вариантах выбора решений другими игроками. В этом разделе рассматривается ситуация, когда оба игрока обладают полной информацией об игре (известны как матрицы выигрышей, так и выигрыши всех игроков, если игра повторяется). Для случая двух игроков оптимальная стратегия состоит в максимизации гарантированного выигрыша:

$$\vec{p}_q^* = \arg \max(\min_{P_C}(\vec{p}_R \cdot M_q \cdot \vec{p}_C)), \quad q = \{R, C\}. \quad (2)$$

Если каждый игрок ищет единственную наилучшую стратегию k_0

$$p_{qk} = \delta_{kk_0(q)}, \quad (3)$$

то полученные решения называются *чистыми* стратегиями.

⁴Традиционно, игрока, выбирающего строку, называют R (row), а игрока, выбирающего столбец — C (column). В дальнейшем символы транспонирования для векторов в формулах будут опускаться.

Решение (2) в чистых стратегиях далеко не всегда существует. Наиболее подробно изучен случай *антагонистических* игр двух игроков (когда сумма выигрышей игроков строго равна нулю — что один выиграл, то другой проиграл). В этом случае

$$M_R = -M_C = M. \quad (4)$$

Если матрица игры имеет седловую точку, то точным результатом, установленным *фон Нейманом*, является равенство цен игр игроков, следующих соответственно, максиминной и минимаксной стратегий:

$$\max_{p_R} (\min_{p_C} (\vec{p}_R \cdot M \cdot \vec{p}_C)) = \min_{p_C} (\max_{p_R} (\vec{p}_R \cdot M \cdot \vec{p}_C)). \quad (5)$$

В этой лекции будет рассматриваться в основном общий случай, когда для трех и более игроков прямого антагонизма нет.

Особенности равновесия в чистых стратегиях иллюстрирует описываемая ниже хрестоматийная стратегическая игра в матричной форме («дилемма заключенного»).

Игра 2. (Дилемма заключенного, ДЗ). Двое задержанных лиц подозреваются в совершении серьезного преступления. Не имея возможности общаться между собой, они поставлены перед выбором, свидетельствовать ли друг против друга или нет? При этом, если один обвинит другого, а тот обвинит первого, то оба получают наказание. Если обвинение будет только со стороны одного из игроков, то он получает свободу, а другой — максимальное наказание. Если же оба откажутся от обвинений (т. е. выберут сотрудничество), то, в силу других обстоятельств дела, они оба получат относительно небольшое наказание. Как поступить рационально?

Несмотря на детективно-развлекательное описание игры, проблема, которая в ней поставлена, является весьма серьезной, а именно, возможно ли возникновение сотрудничества между участниками, преследующими индивидуальные цели? Игра ДЗ имеет более чем полувековую историю, и впервые, по-видимому, обсуждалась в начале 50-х годов в контексте глобальных ядерных стратегий, см. [8]. Она также имеет приложения к проблемам в экономике, политике и биологии. Матрицы игры ДЗ имеют вид, показанный в табл. 1.

Важными являются относительные значения выигрышей игроков. Наибольший выигрыш достигается, если поддаться искушению и обвинить другого игрока. При этом обвиненный игрок окажется в наихудшей из

ТАБЛИЦА 1. Матрицы игры «Дилемма заключенного». Приведены пары выигрышей первого и второго игрока

		<i>Второй игрок</i>	
		Сотрудничество	Обвинение
<i>Первый игрок</i>	Сотрудничество	R, R	S, T
	Обвинение	T, S	P, P

возможных ситуаций. Однако, если оба игрока выбирают стратегию сотрудничества, то выигрыш каждого будет больше, чем в случае взаимных обвинений. Поэтому $T > R > P > S$. Кроме того, оба игрока не должны иметь возможность выйти из дилеммы, по очереди предавая друг друга, т. е. $R > (S + T)/2$. Де-факто обычно используются значения $T = 5, R = 3, P = 1, S = 0$.

Проведем теперь рациональные рассуждения за обоих игроков, используя только чистые стратегии. Предположим, что первый игрок считает, что другой игрок будет придерживаться стратегии сотрудничества. Если первый также будет сотрудничать, то оба получают выигрыши $R = 3$. Если же первый в этих условиях выберет обвинение, то его индивидуальный выигрыш возрастет, и окажется равным $T = 5$. Следовательно, согласно теории рационального выбора, оптимальной стратегией первого игрока, если второй выбрал сотрудничество, будет обвинение.

Пусть теперь первый игрок считает, что второй его обвинит. Если первый в этих условиях выберет сотрудничество (отказ от обвинения), то его выигрыш окажется равным $S = 0$. Однако, если первый также выберет обвинение, то его выигрыш снова возрастет, и окажется равным $P = 1$.

В итоге, при любых предположениях о характере поведения второго игрока, оптимальным для первого игрока будет стратегия отказа от сотрудничества. В этих случаях говорят, что стратегия обвинения *доминирует* над стратегией сотрудничества.

Аналогичной логике следуют рассуждения второго, тоже рационального, игрока. Оптимальные стратегии игроков могут содержаться только среди доминирующих стратегий, и следовательно оба выберут взаимное обвинение, получив при этом выигрыши $P = 1$. Проблемная суть задачи состоит в том, что оба игрока упускают возможность кооперации, которая дала бы им выигрыши $R = 3$. Этот факт и составляет дилемму.

Вопрос о возникновении сотрудничества является крайне важным, и эта игра многократно подвергалась экспериментальной проверке — будут ли реальные люди-игроки всегда следовать равновесию Нэша, являющемуся следствием математической рациональности? Результаты экспериментов [7] варьируются при вариации условий эксперимента (насколько точно удастся воспроизвести ситуацию равенства функций полезности для игроков условиям задачи). Во многих случаях люди чаще (от 50% до 94%) выбирают стратегию обвинения, если не допускается предварительных обсуждений перед ходом в игре. Однако, если оппоненты непосредственно наблюдают друг друга, обвинение возникает реже (от 29% до 70% случаев, в зависимости от фактических условий игры). Мы вернемся к экспериментам ниже, при обсуждении повторяющихся игр.

Равновесия в смешанных стратегиях

Факт обязательного следования игроком чистой стратегии является, в некотором смысле, сковывающим действия игрока, что может быть использовано другими игроками для увеличения их выигрышей. Может ли игрок «скрыть» часть своих намерений и придерживаться разных стратегий в разных актах игры? Ответ на этот вопрос утвердительный. Формальное рассмотрение этого вопроса состоит в введении вероятностей следования игроком той или иной стратегии.

Стратегию, включающую вероятностные комбинации возможных чистых стратегий, называют *смешанной*. Чистые стратегии являются частным случаем смешанных стратегий, когда вероятность одного из выборов равна единице.

Выигрыш игрока при смешанной стратегии p по-прежнему дается выражением (1), в котором под p теперь следует понимать вектор распределения вероятностей чистых стратегий, составляющих смешанную.

Число возможных смешанных стратегий для конечного набора чистых стратегий бесконечно велико. Однако, этот факт, как это часто бывает, лишь упрощает задачу. А именно, для случая смешанных стратегий, имеется фундаментальный результат [4], относящийся к вопросу о существовании равновесия. Напомним, что в случае чистых стратегий установить в общем случае существование состояний равновесия не удавалось. Для игр с конечным набором чистых состояний доказана следующая теорема:

Теорема (Дж. Нэш, 1950).

*Каждая финитная игра имеет точку равновесия*⁵.

Под равновесием в теореме Нэша понимается набор смешанных стратегий таких, что для каждого игрока значение функции полезности (зависящее от его выигрыша при данном наборе стратегий) не может быть увеличено *индивидуальным* уклонением от равновесной смешанной стратегии.

Фундаментальность этой теоремы состоит в том, что она справедлива как для кооперативных, так и для антагонистических игр с любым числом участников. До теоремы Нэша исследования были основаны на классификациях игр с рассмотрением каждого класса в отдельности (именно так построена основополагающая книга [3]).

С практических воззрений теория Нэша не дает ответа на все вопросы. В частности, в теореме ничего не говорится о существовании пути к равновесию в повторяющихся играх и об устойчивости точек равновесия (существовании областей притяжения). Кроме того, в теореме утверждается существование по крайней мере одного равновесия, но, фактически, равновесных точек может быть более одной. Почему игроки будут стремиться в своих стратегиях к одной общей для всех точке? А если это не так, то достижимы ли вообще равновесные состояния в играх? Равновесие по Нэшу имеет смысл только тогда, когда каждый игрок информирован о том, каким стратегиям будут следовать остальные. Если же теоретическое равновесие не единственно, то игроки не будут иметь этой информации.

Вопрос выбора равновесия позднее был положительно разрешен в работах Дж. Харшаньи и Р. Зельтена [5] (разделивших с Дж. Нэшем Нобелевскую премию). Ими была построена общая теория выбора точек равновесия, которая рекомендует единственную⁶ стратегию для каждого игрока. Книга [5] недавно переведена на русский язык и доступна заинтересованному читателю.

Завершая на этом общий экскурс в предмет теории игр хотелось бы остановиться на информационном аспекте, поскольку именно он интересен в контексте тематики Школы по нейроинформатике. Информированность участников об условиях и параметрах игры является основой для

⁵Именно так теорема сформулирована в 27-страничной диссертации Дж. Нэша, воспроизведенной в фотокопии в книге [4]: “ТНЕО. I: Every finite game has an equilibrium point.”

⁶Ложка дегтя заключается в том, что теория Харшаньи-Зельтена о выборе единственного равновесия сама не является единственно возможной. . .

рационального поведения. В современной деловой активности, характеризующейся высоким уровнем конкуренции, свободное распространение информации, существенной для принятия решения, крайне затруднено. При этом особенно охраняется информация о функции полезности игрока, что, конечно, затрудняет выбор даже при известной матрице игры.

Неполнота информации может быть вызвана не только действиями по ее защите, но и большим объемом самой информации. Так, например, участники биржевых игр имеют дело с невероятным по объему потоком новостей от источников с различной степенью достоверности. Разнообразие реакции участников на доступную им информацию определяет общую ценовую ситуацию на рынке. В итоге, наблюдаемые решения рыночных игр весьма разнообразны и динамичны!

Схожая ситуация наблюдается и на рынке консалтинговых услуг, предоставляемых компаниям экспертами или специализированными фирмами. Фактически цена контракта для покупателя определяется его функцией полезности, неизвестной продавцу. Поэтому на практике не существует универсальной «справедливой» стоимости услуг — функции полезности различны у разных компаний, и, кроме того, они могут изменяться во времени.

При неполной информации игроки вынуждены принимать решения на основе своих ожиданий, выраженных в форме информационных *моделей* игры, прогнозирующих как поведение других игроков, так и исход игры. На практике сбор информации для уточнения моделей происходит одновременно с самим процессом игры. К этому вопросу мы и переходим в следующем разделе.

Многошаговые игры

Рассматриваемые ранее стратегические игры со смешанными стратегиями игроков понимались, как повторяющиеся игры с независимыми выигрышами. В каждом конкретном акте игры игрок, конечно, должен был выбрать какой-то конкретный ход. При этом объем информации, доступной каждому игроку, о функциях полезности других игроков и матрице игры предполагался постоянным.

Для моделирования процесса получения и использования в игре информации более реалистичным являются *многошаговые* модели игр.

В многошаговых играх игроки заинтересованы в максимизации сум-

марных выигрышей, полученных на каждом шаге игры. Обычно в качестве целевой рассматривается дисконтированная сумма выигрышей

$$V = \sum_t \gamma^t r(t), \quad (6)$$

где $\gamma < 1$ — фактор дисконтирования, $r(t)$ — выигрыш игрока на шаге t .

Абстракция бесконечного времени игры является существенным отличием от задачи повторяющихся игр с известным конечным числом повторений. Действительно, *последняя* игра в цепочке игр не может повлиять на сложившиеся исходы предыдущих актов, и следовательно, должна рационально рассматриваться игроками как обычная одношаговая стратегическая игра. Это одинаково понимается всеми рациональными игроками, и, следовательно, задача сводится к цепочке из $(n - 1)$ игр. Это рассуждение по индукции приводит к рассмотрению каждой игры из повторяющейся серии как игры в независимой стратегической постановке.

В многошаговой игре с дисконтированным суммарным выигрышем стратегия нетривиальным образом зависит от исходов многих игр⁷.

Игра «Дилемма заключенного», описанная в предыдущем разделе, экспериментально исследовалась и в многошаговой постановке [8], при этом в качестве игроков выступали компьютерные программы разных авторов. Турниры между попарно играющими программами были организованы Робертом Аксельродом (Robert Axelrod) в 1980 году и позднее повторены спустя несколько лет. На основе стратегий, предложенных в программах, была сформулирована смешанная стратегия, вероятности в которой определялись генетическим оптимизационным алгоритмом. В итоге, в популяции стратегий выживали, в основном, стратегии типа «око-за-око», в которых игрок принимает решение сотрудничества в первой игре, а далее повторяет ходы оппонента. Однако ненулевые вероятности имели также и другие стратегии, например «обидчивая» стратегия, в которой игрок сотрудничает до первого отказа от сотрудничества со стороны оппонента, а далее всегда выбирает решение обвинения. Подробная информация об этой игре и турнирах Аксельрода имеется в Интернете [16].

⁷В этом смысле многошаговая игра может рассматриваться как специальный вариант известной задачи об останове [11]. В игре участвует еще один игрок, ходом которого является вероятностное решение, продолжать дальше многошаговую игру или нет. Если игрокам известна вероятность продолжения?, то, с их точки зрения, полученная игра эквивалентна многошаговой игре с фактором дисконтирования γ .

Поиск оптимальной стратегии в многошаговой игре может быть представлен как обобщение формализма марковского процесса решений (Markov Decision Process) на случай нескольких игроков.

Для одного игрока (агента) задача оптимизации поведения была подробно рассмотрена в предыдущей лекции автора [12], поэтому приведем лишь основные соотношения, которые и будут обобщены. Марковский процесс принятия решений представляет собой совокупность из множества состояний среды (игры!) $s \in S$, множества возможных решений агента $a \in A$, функции подкрепления (выигрыша!) агента $r(s, a)$, а также плотности вероятности $T(s'|s, a)$ переходов между состояниями среды.

Выбор решений агента определяется его стратегией $p(a)$, заданной как плотность распределения на множестве решений. Оптимальная стратегия определяется уравнением Беллмана (см. список литературы в [12]):

$$V(s, p^*) = \max_a \left\{ r(s, a) + \gamma \sum_{s'} T(s'|s, a) \cdot V(s', p^*) \right\}, \quad (7)$$

где V — «цена» посещения состояния s . Для оптимизации выбора стратегии удобнее перейти к понятию цены Q пары $\langle \text{состояние, решение} \rangle$, которая равна суммарному дисконтированному подкреплению при принятии в состоянии s решения a и следовании оптимальной стратегии в дальнейшем:

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a'). \quad (8)$$

Для практических вычислений широко применяются итерационное Q -обучение и алгоритм SARSA, в сочетании с ε -оптимальным поведением агента на оптимизируемой траектории.

В теоретико-игровой постановке цены игр V зависят от решений нескольких игроков:

$$V_q(s, p_q^*; \vec{p}_{-q}^*) = \sum_t \gamma^t \langle r_q(s(t), p_q^*; \vec{p}_{-q}^*) | s_0 = S \rangle. \quad (9)$$

В формулах индексом $(-q)$ обозначены наборы решений всех игроков кроме данного, угловые скобки указывают на то, что r является решением соответствующей стратегической игры на каждом шаге. Каждый игрок при поиске оптимальной стратегии должен был бы просто максимизировать свою функцию ценности Q в новом состоянии S' :

$$Q_q(s, a_q; \vec{a}_{-q}) \stackrel{?}{=} r_q(s, a_q; \vec{a}_{-q}) + \gamma \max_{a'} Q_q(s, a'_q; \vec{a}_{-q}), \quad (10)$$

однако непосредственная индивидуальная ее максимизация невозможна, так как функция Q зависит от решений других игроков. Для выхода из противоречия [13] заметим, что $Q_q(s, a_q; \vec{a}_{-q})$ определяет матрицу некоторой игры в состоянии s' и оптимальная стратегия опирается на смешанную стратегию $p(a'_q)$ в этой игре.

В частном случае строго антагонистической игры агент выбирает максимизацию гарантированного выигрыша:

$$Q_q(s, a_q; \vec{a}_{-q}) = r_q(s, a_q; \vec{a}_{-q}) + \gamma \max_{p(a'_q)} \min_{p(a'_{-q})} Q_q(s, a'_q; \vec{a}_{-q}). \quad (11)$$

В общем случае второе слагаемое в сумме (11) содержит решение матричной игры

$$\{Q_q(s, a'_q; \vec{a}_{-q}); Q_{-q}(s, a'_q; \vec{a}_{-q})\},$$

которое учитывает игровые матрицы всех игроков:

$$Q_q(s, a_q; \vec{a}_{-q}) = r_q(s, a_q; \vec{a}_{-q}) + \gamma \max_{p(a'_q)} \min_{p(a'_{-q})} \langle p'_q \cdot Q_q(s, a'_q; \vec{a}_{-q}) \cdot p_{-q} \rangle. \quad (12)$$

Здесь по-прежнему угловыми скобками обозначена цена игры.

Решение (12) отражает характер получения информации при взаимодействии агента с игровой средой: для поиска оптимальной стратегии каждый агент кроме формирования собственной функции цены состояния Q_q должен обучаться моделям матриц выигрышей Q_q всех остальных игроков. При этом их текущие выигрыши должны быть ему известны на каждом шаге (хотя фактические смешанные стратегии других игроков и их фактические матрицы игр остаются скрытыми). В каждый момент времени агент оптимизирует свое поведение в текущем равновесии Нэша для текущей модели игры. Это еще раз подчеркивает отличие стохастических многошаговых игр от повторений одной стратегической игры: в многошаговой игре матрицы выигрышей Q эволюционируют во времени.

Для получения оптимальных стратегий, также как и в случае обычного обучения с подкреплением, агент может следовать ε -оптимальной стратегии на каждом шаге, с вероятностью ε уклоняясь от текущего оптимального («жадного») распределения вероятностей своих ходов с целью исследования всего пространства решений.

Сходимость получаемых стратегий в модели типа (12) исследовалась в работах авторов [12, 16] и установлена для ряда специальных предположений о характере игр. Однако вопросы совместного выбора общего равновесия остаются открытыми. Нужно заметить, что полученные модели

игр не обязательно сходятся к фактическим матрицам выигрышей, а модели стратегий — к фактическим стратегиям игроков-соперников данного агента.

Вычислительные затраты на реализацию модели (12) значительны. Обучаемые параметры включают в себя матричные элементы всех матриц игр всех игроков, при этом теоретически сходимость обучения гарантирована только для случая, когда каждое из состояний игры посещается (бесконечно) часто [16]. Для вычисления итерационных поправок к ценам Q требуется на каждом шаге решать серию задач квадратичного программирования для определения векторов вероятностей в смешанных стратегиях каждого игрока. Прямой путь, как и в случае обучения с подкреплением, состоит в использовании нейросетевых аппроксимаций для всех Q -факторов, при этом в качестве оценки ошибки аппроксимации используются разности старых и новых значений Q -факторов на временном шаге (TD -алгоритм, [14]).

Для практических целей в нашей лекции будет рассмотрена упрощенная постановка задачи обучения агента-игрока, в которой неопределенность в ценах игры, вызванная решениями других игроков, моделируется статистически. Одновременно с упрощением вычислений предлагаемая модель является более общей, так как она включает случай, когда агенту известен только его текущий выигрыш в игре, а значения выигрышей других игроков не известны. Полная теория игровых моделей такого типа далека от завершения, поэтому в лекции обсуждается на практическом примере из области организации работы холдинга компаний, оказывающих схожие услуги на общем рынке. Пример иллюстрирует общие проблемы, возникающие при игровом моделировании, а также характер результатов, которые можно получить из игровых моделей.

Описание основного вычислительного звена модели — нейросетевой архитектуры, аппроксимирующей стохастическое значение цены игры, вынесено в Приложение.

Модель многошаговой игры с неполной информацией

Рассмотрим ситуацию, когда игрок вынужден принимать последовательность решений в многошаговой игре в условиях, когда матрицы игры и выигрыши других игроков ему не известны. Фактически, в этих условиях неизвестно даже общее число игроков, участвующих в игре. Постановка

такой задачи весьма близка к реальной ситуации при игре на бирже или при деловой деятельности в условиях конкуренции на рынке.

До начала игры игроку⁸ известно множество ходов, которые он может делать⁹ в игре. Единственная информация, которой дополнительно снабжается игрок в течение игры — это значения его выигрышей. Игроку также известен его текущий счет. В чуть более ослабленной постановке игроку может сообщаться информация о значениях некоторых переменных, характеризующих текущее состояние игры при играх с несколькими состояниями. Эти переменные добавляются к переменным собственного состояния агента (например, его текущий счет, пространственное расположение и т. п. определяются конкретными особенностями игры).

Целью агента по-прежнему является максимизация собственного дисконтированного выигрыша в последовательности игр. Q -фактор агента (максимальное значение суммарного выигрыша при начале игры из состояния s , в котором агент принимает решение a и далее следует избранной стратегии) зависит от «скрытых» переменных — решений других игроков. Поэтому непосредственное использование итерационных алгоритмов (12) затруднено.

Эффект скрытых переменных, который невозможно при отсутствии информации учесть точно, может быть описан статистически. В предлагаемой вероятностной модели значение Q -фактора объявляется случайной величиной с параметрической плотностью распределения, моменты которого зависят только от переменных состояния агента и его решений. Без ограничения общности будем считать параметрическое распределение *при каждом наборе переменных* нормальным (с двумя функциональными параметрами — математическим ожиданием и дисперсией, зависящими от переменных). Обобщение на случай других распределений, как будет видно из дальнейшего, состоит в использовании при обучении других видов функции правдоподобия. Гипотезу о соответствии распределения выбранной форме в каждой практической задаче следует проверить статистическими критериями. Описание соответствующих процедур имеется в стандартных курсах статистики, в данном тексте эти вопросы не затрагиваются.

⁸В этой лекции мы прежде всего интересуемся вопросами принятия решений игроками-компьютерными программами, поэтому в тексте не делается различий между терминами *игрок* и *агент*.

⁹В действительности, это требование может быть ослаблено, если игрок будет получать отрицательные выигрыши при невозможных ходах. В этом случае фактические границы области допустимых ходов устанавливаются агентом в процессе обучения в игре.

Собственно, специфика нормальности распределения никак не используется¹⁰, кроме явного вида самой функции и ее производных.

Вероятностная модель функции цены пары $\langle \text{состояние}, \text{решение} \rangle$ имеет вид:

$$Q_q(s, a_q; \vec{a}_{-q}) \propto \tilde{Q}(s, a_q) = N(m(s, a_q), \sigma(s, a_q)). \quad (13)$$

Моменты гауссовой плотности распределения N рассматриваются как независимые функции (от известных переменных), подлежащие определению в процессе обучения. Их удобно представить в виде нейросетевых аппроксимаций. На практике для аппроксимации может использоваться одна и та же нейронная сеть, но с двумя выходами.

Теперь необходимо обобщить понятие ожидаемого выигрыша в игре с вероятностной матрицей \tilde{Q} . Пусть имеется некоторая j -я реализация значений матрицы в состоянии s для (дискретного) набора их k возможных решений агента

$$\tilde{Q}_j(s, \vec{a}) = [\tilde{Q}(s, a_1), \tilde{Q}(s, a_2), \dots, \tilde{Q}(s, a_k)]_j. \quad (14)$$

Ожидаемым исходом j -й реализации игры для агента со смешанной стратегией $p(a)$ является величина

$$\sum_k p(a_k) \cdot Q_j(s, a_k). \quad (15)$$

Наилучшим ходом в этой реализации игры будет

$$a_j^* = \arg \max_k Q_j(s, a_k). \quad (16)$$

Наилучшей «жадной» стратегией в целом является (выборочный) закон распределения величин $\{a_j^*\}$, который может оцениваться методом Монте-Карло, как статистика¹¹ большого числа выборок \tilde{Q}_j . Уравнение обучения с подкреплением в форме временных разностей имеет вид:

$$\Delta \tilde{Q}(s, a) = [(s, a) + \gamma \langle \max_{a'} \tilde{Q}_j(s', a') \rangle - \langle \tilde{Q}(s, a) \rangle] \quad (17)$$

¹⁰Предпочтительнее использовать распределения, определяемые небольшим числом достаточных статистик, и допускающие эффективную генерацию выборок для метода Монте-Карло.

¹¹Напомним, что здесь рассматривается конечное множество возможных решений a_k .

(угловыми скобками обозначено усреднение по реализациям игры). Это уравнение является специальным видом рекуррентного уравнения Беллмана для поиска оптимума аддитивных функций. Колебания величины (17) определяют дисперсию плотности вероятности величины \tilde{Q} .

Завершающим шагом описания модели адаптации агента в многошаговой игре является вывод соотношений для построения нейросетевых аппроксимаций среднего и дисперсии распределения \tilde{Q} .

Аппроксимация плотности проводится путем максимизации функции правдоподобия (maximum likelihood). Выбор нейросетей в качестве пробных функций продиктован их хорошими аппроксимационными свойствами и возможностью быстрого вычисления градиента по свободным весовым параметрам при решении задачи максимизации.

В этой работе использовалась эффективная нейронная сеть локальных нормализованных сплайнов (Connectionist Normalized Local Splines — CNLS), предложенная в начале 90-х годов специалистами Центра исследований нелинейности¹² в Лос-Аламосской национальной лаборатории [18,19]. Важной особенностью этой нейросети является не только возможность быстрого обучения, но и специфические свойства базисных функций, дающих близкую к кусочно-линейной аппроксимацию с автоматической гладкой «сшивкой» областей. Публикации по применениям этой сети немногочисленны, что, по-видимому, сдерживает ее широкое распространение¹³. Описание функционирования и обучения нейросети CNLS представляет собой самостоятельную тему и поэтому вынесено в Приложение.

Искомые функциональные зависимости $m(s, a)$ и $\log \sigma(s, a)$ представляются двумя выходами нейронной сети, входами которой является совокупность переменных состояния (их может быть несколько) и переменной, описывающей решение агента. Будем считать, что все переменные представлены вещественными числами из некоторой ограниченной области определения. Вместо максимизации функции правдоподобия удобнее минимизировать ее логарифм, имеющий для гауссового распределения вид:

$$-\log L = \frac{1}{2}(2\pi) + \log(\sigma(s, a; w)) + \frac{(m(s, a; v) - F(s, a))^2}{2\sigma(s, a; w)^2}, \quad (18)$$

¹²Аббревиатура названия Центра — CNLS (Center for Non-Linear Studies).

¹³Этот факт уже несколько лет приводит автора в недоумение. Нейронная сеть этой архитектуры значительно превосходит и по скорости и по удобству использования популярные нейросети с обратным распространением ошибки (backpropagation). В лекции предоставился удобный случай подробно описать эту нейроархитектуру.

где $F(s, a)$ — наблюдаемое значение аппроксимируемой величины, известное в традиционной задаче обучения нейросети с учителем и *неизвестное* в рассматриваемом здесь случае обучения с подкреплением. На текущем шаге обучения разность значений $(m - F)$ заменяется ее оценкой (17), являющейся разностной производной аппроксимируемой функции по времени.

Обе пробные функции зависят от наборов весовых коэффициентов нейросетевой модели (w и v). Обучение основывается на значении градиента¹⁴ минимизируемого функционала по этим параметрам. Непосредственные вычисления дают:

$$\frac{\partial(-\log L)}{\partial v_i} = \frac{\Delta Q(s, a)}{\sigma(s, a; w)^2} \cdot \frac{\partial m(s, a; v)}{\partial v_i}, \quad (19)$$

$$\frac{\partial(-\log L)}{\partial w_j} = \left[1 - \left(\frac{\Delta Q(s, a)}{\sigma(s, a; w)} \right)^2 \right] \cdot \frac{\partial \log \sigma(s, a; w)}{\partial w_j}. \quad (20)$$

Выражения (19)–(20) справедливы для дифференцируемых нейросетей любой архитектуры (без рекуррентных связей). Специфика конкретного типа нейросети проявляется в конкретном виде производных выходов по параметрам w и v .

Заметим, что большинство реализаций алгоритмов обучения игнорируют зависимость дисперсии плотности вероятности, даваемую соотношением (20), довольствуясь лишь аппроксимацией математического ожидания (19). При постоянной дисперсии ее значение не сказывается на направлении градиента и знаменатель в (19) полагают равным единице. В этом пределе формула (18) переходит в метод наименьших квадратов. Максимизация функции правдоподобия с аппроксимацией всех параметров выбранного распределения является более универсальным (и численно более устойчивым) подходом. Соотношения вида (19)–(20) могут быть легко выведены и для других (отличных от гауссового) законов распределения вероятности. Это становится важным, например, если фактическое распределение уклонений от математического ожидания является несимметричным или имеет широкие крылья (что часто встречается в финансовых приложениях

¹⁴При обучении целесообразно использовать более мощные методы оптимизации, чем прямой метод стохастического градиента. Литература по применениям методов оптимизации к обучению нейронных сетей весьма обширна, и читатель без труда найдет необходимые книги и статьи. Методы оптимизации реализованы в большинстве свободно распространяемых и коммерческих компьютерных программ, моделирующих нейронные сети.

и при прогнозировании временных рядов). Большим преимуществом метода является одновременное построение и самой аппроксимации, и оценки ее ошибки в каждой точке, определяемое функциональной зависимостью дисперсии от переменных задачи.

Итак, в этом разделе сформулирован вычислительный подход к нахождению оптимальных смешанных стратегий при принятии решений в многошаговых играх. Основным его содержанием является представление цены стратегии в текущем состоянии игры в параметрической вероятностной форме. Параметры распределения определяются уравнением Беллмана, решения которого находятся в виде нейросетевых аппроксимаций.

Завершающая часть лекции посвящена применению предложенного подхода к решению практической задачи теоретико-игрового моделирования и оптимизации бизнес-процессов на внутреннем рынке холдинга консалтинговых компаний.

Задача о деятельности холдинга консалтинговых компаний

Фундаментальной областью приложений теории игр является проблема коллективного поведения и взаимодействия игроков-участников совместной бизнес-деятельности. Проблема весьма многогранна и в этой лекции мы ограничимся подробным рассмотрением конкретного примера из области организации деловой активности. В качестве иллюстрации выбрана деятельность холдинга компаний, специализирующихся на предоставлении консалтинговых услуг в нескольких смежных областях, относящихся к информационным технологиям. В этой главе предложена *модель* внутренней конкуренции и кооперации, основанная на аукционных механизмах выработки управляющих решений.

Основу модели составляют нейросетевые методы выбора стратегии рационального поведения для индивидуальной компании-участника холдинга. При этом игровые условия учитывают неполноту и *вероятностный* характер информации, доступной каждому игроку.

Модель бизнес-процессов

Несколько компаний оказывают консалтинговые услуги в области информационных технологий. Экспертные профили компаний близки по темати-

кам выполняемых проектов и с целью снижения маркетинговых и других издержек компании объединяются в холдинг, управляемый головной компанией, не занятой в проектах непосредственно. Головная компания имеет фиксированную долю прибыли от каждого успешно проведенного проекта.

Имеется фиксированный набор тематических разделов, однако каждый проект может потенциально содержать задачи из разных разделов. Профиль проекта задается распределением представленных в нем тематик. Стоимость каждого проекта для внешних заказчиков фиксирована (и равна 1). Проект полностью выполняется одной из компаний холдинга.

Каждая компания имеет свой профиль распределения занятых в ней экспертов по тематикам. Этот профиль представлен в виде вероятностей успешного выполнения проекта по каждой из тематик в отдельности. Таким образом, экспертный уровень компании (вероятность успешного выполнения комплексного проекта) равен скалярному произведению профиля проекта и профиля компании. Предполагается, что компании имеют достаточно высокий экспертный уровень, однако их экспертизы по отдельным тематикам разнятся.

Каждая компания использует свой собственный набор методик и программного обеспечения, а фактические экспертные уровни остальных компаний ей неизвестны. Неизвестными являются также и функции полезности других компаний, определяемые только их уровнем текущих доходов в расчете на один проект. Состояния счетов не публикуются.

При успешном выполнении проекта его финансовое обеспечение, за вычетом доли управляющей компании, целиком поступает на счета компаний. Если проект завершается неудачно, то холдинг несет убытки в размере полной стоимости проекта плюс дополнительные издержки, процент которых также фиксирован. Убытки не затрагивают управляющую компанию.

Компании холдинга в целом остаются независимыми и преследуют, прежде всего, интересы своего собственного бизнеса. Для организации совместной деятельности в холдинге предложена схема деятельности, основанная на внутреннем аукционе. Рациональность этой схемы, а также оптимальная доля прибыли головной компании и являются предметом исследования.

Схема бизнес-процессов состоит в следующей последовательности этапов, составляющих один цикл (временной шаг).

1. На текущем шаге холдинг получает заказ на выполнение нового проекта, случайно выбираемого из фиксированного списка проектов. Профиль тематик проекта сообщается всем компаниям-участникам.

2. Каждая компания индивидуально вычисляет свой собственный экспертный уровень относительно данного проекта и владеет информацией о состоянии своего счета.
3. Компании делают одинаковый страховой взнос, суммарная по компаниям величина которого равна стоимости проекта и фиксированных издержек, которые возникнут, если выполнение проекта будет сорвано.
4. Каждая компания назначает свою, внутреннюю для холдинга, цену выполнения проекта (не превышающую стоимость проекта от заказчика). Заявленные цены передаются в управляющую компанию и держатся в секрете от других участников.
5. Заявки компаний участвуют во внутреннем аукционе, при этом побеждает заявка с минимальной ценой.
6. Если цена победителя не превышает отпускную цену проекта за вычетом фиксированной доли управляющей компании, то компания-победитель аукциона получает возможность выполнить этот проект *по заявленной ею цене*. В противном случае холдинг просто отказывается от выполнения этого проекта, не получая на этом шаге ни доходов, ни убытков. Страховые взносы полностью возвращаются компаниям.
7. С вероятностью, равной своему экспертному уровню относительно данного проекта, компания-победитель аукциона успешно выполняет проект. В этом случае она получает доход, равный заявленной ею аукционной цене, управляющая компания получает фиксированную долю, а остаток средств равномерно распределяется по остальным компаниям холдинга. Страховые взносы полностью возвращаются компаниям. Таким образом, все заинтересованы в успешном выполнении проекта.
8. Если же проект окажется неуспешным, то компании теряют свои страховые взносы, управляющая компания ничего не получает.

После этапа 8 происходит новый цикл деятельности.

Таким образом, компании поровну делят риски совместной деятельности, а также частично и доходы от нее. Управляющая компания, обеспечивающая приток новых проектов, потенциально имеет единственный рычаг долгосрочного управления — ставку своей прибыли. В течение моделируемого периода времени эта ставка остается постоянной.

Каждая компания также имеет одну степень свободы — внутреннюю цену каждого проекта, участвующую в аукционе. Прямой путь повышения собственных доходов — увеличение этой цены, однако, высокие заявки имеют малый шанс победить в аукционе. При низких же ценах компании самой не выгодно выполнять проект — пусть это сделает кто-то другой, а она получит больше при дележе остатка денег. С другой стороны, передача проекта в компанию, имеющую в его тематиках невысокий экспертный уровень, также невыгодна — возрастает риск неудачи проекта. Потери равномерно распределяются между участниками, но это не влияет на выбор компаний, так как они не преследуют цель относительной успешности на рынке, важен лишь собственный уровень доходов в расчете на один заказанный холдингу проект. Высокие цены на проект со стороны всех компаний приведут к тому, что проект вообще не будет выполняться (но время упущено и это учитывается в подсчете числа заказанных проектов).

Целью каждой компании является нахождение стратегии выбора цен в аукционах, максимизирующей суммарный (дисконтированный) доход.

Формализация математической модели многошаговой игры

Эта схема, с математической точки зрения, представляет собой многошаговую игру с несколькими игроками, обладающими неполной информацией. Выигрыши других игроков, их функции полезности и матрицы игры данному игроку неизвестны. Фактически ему известна только тематика серии проектов, состояние собственного счета и значение его выигрыша (с учетом страхового взноса) на каждом шаге.

Принцип моделирования такой игры был предложен в предыдущем разделе лекции. Для его детализации введем обозначения:

$j(t)$	Номер состояния игры на шаге t .
	Состояние игры задается типом (распределением тематик) нового проекта. Число типов конечно.
n	Число игроков.
e_{qj}	Экспертный уровень компании q в тематиках проекта j . Равен вероятности успешного выполнения проекта этой компанией. Экспертный уровень вычисляется скалярным произведением

$u_q(t) = U_q(t)/t$	профиля компании и профиля проекта. Каждой компании известен только ее экспертный уровень.
$h(t) = H(t)/t$	Относительный доход компании в расчете на один проект (U — текущее значение счета компании).
$s_q(t) = [u_q(t), e_q(t) = e_{qj}(t)$	Относительный доход управляющей компании холдинга.
$a_q(t)$	Совокупный вектор состояния агента-компании на шаге t .
$p_q(t)$	Решение агента (фактическая заявленная ставка на аукционе) на шаге t .
$\tilde{Q}_q^p(s, a)$	Смешанная стратегия (распределение аукционной ставки) компании q на шаге t . Определяется в виде кусочно-постоянной функции на заданном конечном разбиении диапазона возможных ставок $[0..1]$.
$m_q^p(s, a), \sigma_q^p(s, a)$	Q -фактор агента при стратегии p (случайная функция с локально нормальным распределением значений при каждом выборе аргументов). Математическое ожидание Q равно суммарному дисконтированному выигрышу агента при начале игры из состояния s , в котором агент принимает решение a и далее следует выбранной стратегии p .
$\tilde{Q}_q^*(s, a)$	Среднее и дисперсия распределения значений Q -фактора.
γ	Q -фактор при оптимальной стратегии p^* .
ξ	Фактор дисконтирования.
	Фиксированная доля дохода управляющей компании в каждом

η	успешном проекте. Фиксированный процент дополнительных издержек (сверх суммы проекта) при его неудаче.
$b(t)$	Значение заявки, выигравшей аукцион (равно минимальной заявке, сделанной участниками).
$r_q(t)$	Фактический выигрыш агента на данном шаге игры (может быть отрицательным при неудачном проекте).

Каждый агент после получения информации о своем выигрыше на шаге выполняет итерацию обучения своего Q -фактора цен состояний.

Прежде чем описать алгоритм математической модели игры, отметим, что в условиях задачи одна или несколько фирм могут разориться (получить отрицательное значение счета) на ранних шагах игры. Фактически, счета фирм испытывают несимметричные случайные блуждания¹⁵ с вероятностью шага вверх, равной экспертному уровню e фирмы в проекте, а также соответствующей вероятности шага вниз, равной $(1 - e)$. Эта задача сводится к известной вероятностной задаче о разорении, рассмотренной подробно в курсах теории вероятностей (например, в фундаментальном труде *Феллера*). Экспертные уровни предполагаются достаточно высокими и в контексте данной лекции задача о разорении не рассматривается. Игровые траектории, приводящие к разорению хотя бы одной фирмы, просто исключаются из общей статистики игр.

Рассматриваемая многошаговая игра проводится согласно алгоритму, приведенному на с. 118.

Индивидуальные стратегии игроков

Прежде чем приступить к моделированию, обсудим особенности решений и равновесий, которые могут возникать в данной модели, в случае, когда игроки *обладают информацией об экспертном уровне друг друга*¹⁶.

¹⁵В действительности, это «управляемые» блуждания, поэтому точное решение задачи о вероятности нулевого счета затруднено.

¹⁶В противном случае достижение равновесия рациональным логическим путем невозможно.

Алгоритм 1: Алгоритм многошаговой игры

```

for цикл по раундам игр do
   $t := 0$ ;
  инициализация:  $\vec{u} := 0$ ;  $h := 0$ ;  $\tilde{q} := \tilde{Q}_0$ ;
  for цикл по шагам игры do
     $t := t + 1$ 
    разыграть тематику нового проекта  $j(t)$ ;
    for all для каждого игрока  $q$  do
      определить состояние  $s_q := [u_q, e_q]$ ;
      найти  $\varepsilon$ -«жадную» стратегию  $p_q(t)$ ;
      разыграть ставку  $a_q(t)$  из распределения  $p_q(t)$ ;
    end for
    провести аукцион  $b(t) := \min_q \{a_q\}$ ;
    провести розыгрыш успешности проекта для победителя;
    if проект успешен then
      обновить счет управляющей компании  $H := H + \xi$ ;
      найти выигрыш  $r_q(t)$ ;  $U_q := U_q + r_q$ ;
    else
      for all для каждого игрока  $q$  do
        вычесть страховой взнос  $U_q := U_q - (1 + \eta)/\eta$ ;
      end for
      for all для каждого игрока  $q$  do
        провести цикл обучения  $\tilde{Q}_q$ ;
      end for
      накопить статистику шага игры;
    end if
  end for
  накопить статистику раунда игры;
end for
  стоп;

```

Для числовых оценок примем конкретный набор параметров игры, этот же набор параметров будет в дальнейшем использован в компьютерном моделировании:

$$n = 3; 0.7 < e < 1; \gamma = 0.99; \xi = 0.1; \eta = 0.1.$$

Рассмотрим зависимости математического ожидания выигрыша игрока от сложившейся аукционной цены для двух случаев: когда победителем аукциона является этот игрок и когда проект выполнялся каким-то другим игроком.

Важной точкой на кривой выигрышей является точка безразличия (6), в которой доход не зависит от того, кто фактически победил в аукционе — данный игрок или какой-то другой игрок. На кривой выигрыша имеется $(n - 1)$ таких точек — для каждого из оппонентов. Чем сильнее оппонент, тем правее расположена эта точка (напомним, что у сильных оппонентов выше вероятность успешного завершения проекта).

Соотношения для координат точек легко получить прямо из условия задачи. Математические ожидания складываются из вероятности успеха проекта с соответствующим дележом выигрышей и вероятности его провала, который штрафует.

Для справки, точка «безразличия» (№ 6) некоторого игрока q по отношению к игроку p имеет абсциссу:

$$\frac{(e_p - e_q) \cdot (1 + \eta) \cdot \frac{n-1}{n} + e_p \cdot (1 - \xi)}{e_q \cdot (n - 1) + e_p}. \quad (21)$$

Особенность игры состоит в том, что игрок-победитель может влиять на цену аукциона только понижая свою ставку, что, конечно, выгодно всем другим игрокам. Если игроки обладают полной информацией об экспертном опыте оппонентов, то рациональным исходом аукциона в чистых стратегиях будет цена безразличия между самым сильным и самым слабым игроком на диаграмме победителя (диаграммы разных игроков отличаются наклонами линий, пропорциональными их экспертному опыту). Все, кроме победителя, заинтересованы в понижении этой цены, победитель же предпочел бы ее повысить (но остаться при этом победителем! — это и сдерживает повышение).

Заметим, что эта точка находится ниже «всеобщей средней» цены. Если такая ситуация имеет место для всех типов выполняемых проектов, то чистое равновесное состояние в каждом отдельном шаге игры становится

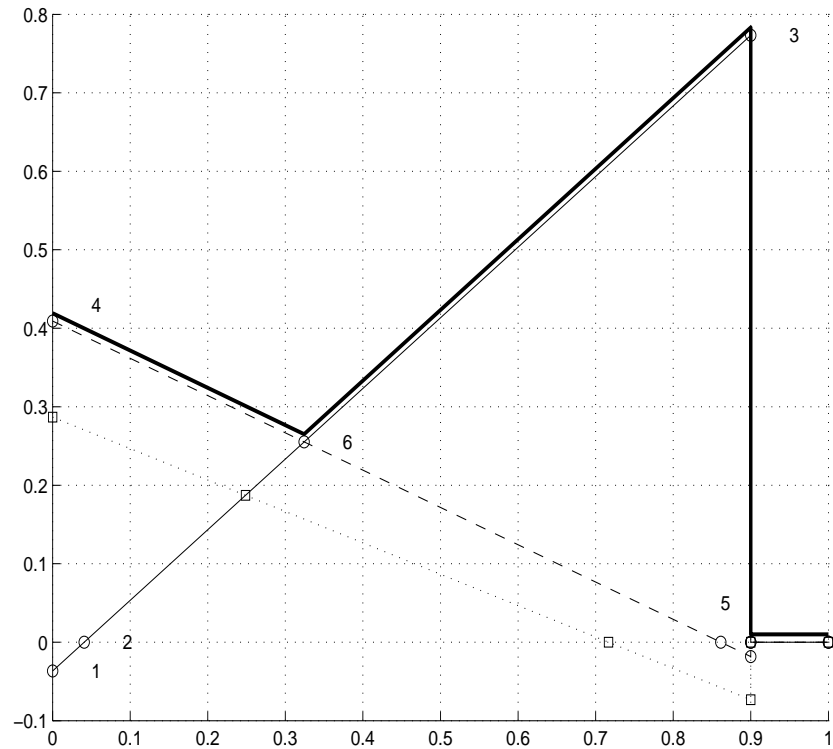


Рис. 1. Зависимость ожидаемого выигрыша игрока от результирующей цены на аукционе

Сплошная тонкая линия — выигрыш, если победила ставка этого игрока. *Прерывистая линия* — выигрыш, если победил игрок, экспертный опыт которого выше, чем у данного игрока. *Тонкий пунктир* — выигрыш, если победил игрок с меньшим экспертным опытом. *Сплошная жирная линия* — наиболее предпочтительный сценарий для данного игрока. Отмечены точки выигрышей при нулевой цене (1,4), точки безубыточности (2,5), максимально возможный выигрыш (3) и точка «безразличия» (6).

невыгодным самому сильному игроку — он выполняет *все* проекты, но имеет доход ниже среднего. Поэтому он будет вынужден искать смешанную стратегию, в которой время от времени побеждает второй по силе игрок. Низкие защитные заявки слабых игроков в этом случае будут приводить к их победам на аукционе, но достигаемый при этом выигрыш ниже, чем при уступке проекта более сильным игрокам. Таким образом, в многошаговой игре равновесие Нэша в чистых стратегиях для одной стратегической игры (играемой на каждом шаге) неустойчиво.

Видно, что картина многошаговой игры достаточно сложна даже в случае полной информации. Нас же будет интересовать практически важный случай, когда игроки не обладают полной информацией (в частности, не имеют значений экспертного опыта оппонентов). Рассмотрение такой игры может быть проведено, по-видимому, только в компьютерной модели.

Компьютерная система COGITO

Описанный алгоритм проведения игры реализован в компьютерной системе COGITO. Система позволяет описывать переходы между состояниями многошаговой игры, проводить аукционы, вычислять выигрыши игроков. Модель игрока основывается на предложенном нейросетевом алгоритме аппроксимации плотности распределения его цены состояния, она позволяет проводить обучение нейросети с подкреплением и оценивать текущую оптимальную смешанную стратегию.

Система COGITO позволяет изменять механизмы аукционов (например, в коммерческих аукционах в качестве цены часто выбирается цена второй заявки, следующей за ценой победителя), а также использовать алгоритмы распределения, отличные от аукционов (например, комитетные решения). В COGITO может также «дозироваться» информация, сообщаемая игрокам, например, могут публиковаться выигрыши и уровни экспертизы всех других игроков или их части.

Статистическая информация, собираемая в системе, включает все аспекты каждого шага игры (ставки игроков, цену аукциона, выигрыши и текущие счета игроков).

Важно также, что на аукционные предпочтения игроков влияет ставка ξ прибыли управляющей компании. При высоких ставках прибыли в выполнении проектов включаются более слабые компании, это приводит к понижению вероятности успешности проектов и, как следствие, к снижению общих доходов управляющей компании. Модель игры позволяет

определить *оптимальные значения норм прибыли*, а также выяснить, как они зависят от процента издержек при неудаче в проекте. Таким образом, компьютерная игровая система позволяет провести оптимизацию деятельности управляющей компании.

Выборочные результаты моделирования

Система COGITO позволяет получать решения игр и смешанные стратегии игроков в различных вариантах многошаговых игр. С практической стороны интересны возможные типы получаемых результатов. Некоторые из них проиллюстрированы ниже на рисунках. В приведенных иллюстрациях с целью экономии времени вычислений игра проводилась для случая трех игроков (не считая управляющей компании, параметры которой предполагались постоянными) и двух типов проектов по двум возможным тематикам с профилями $[0,1]$ и $[1,0]$. Профили игроков были устроены таким образом, что два из них были безусловными экспертными лидерами (каждый по своему направлению), а третий игрок имел высокие, но уступающие лидерам, экспертные уровни в каждом из проектов. Выбраны $e_1 = [0.8, 0.8]$, $e_2 = [0.95, 0.75]$, $e_3 = [0.71, 0.99]$.

Первый интересный вопрос состоит в оценке относительного успеха первого и второго игроков. Причина — суммарная экспертиза первого игрока равна 1.6, что меньше значения 1.7 у второго и третьего участника. Однако первый не имеет очевидных «слабостей». Что в данных условиях лучше?

Интересно также посмотреть, как распределения цен игроков сходятся во времени.

И в заключение этого параграфа приведем вероятности выигрыша каждым из игроков в каждом из типов проектов в течение всего набора игр (см. табл. 2).

Играющий чаще — проигрывает!

Обсуждение

Первое наблюдение состоит в том, что в условиях крайне ограниченной информации фирмы-участницы находят способ рационального позиционирования на внутреннем рынке. Равновесные стратегии имеют интерпретируемые распределения. Смешанные стратегии сильных и слабых игроков

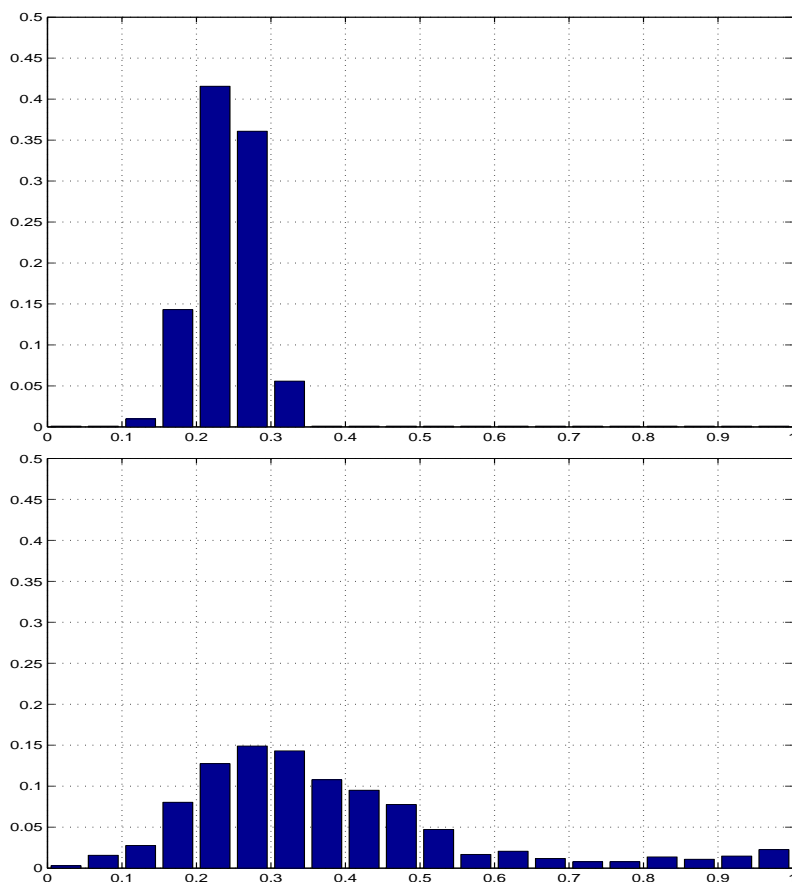


Рис. 2. Смешанные стратегии сильного третьего (верхний рисунок) и «слабого» первого игроков в отношении одного из проектов на поздних стадиях многошаговой игры

Слабый игрок с некоторой вероятностью делает заявки в области точки безразличия сильного игрока, цены слабого игрока в целом выше — при риске стать победителем он предпочитает более высокие ставки. Совсем низкие ставки (с вероятностью около 0.18) у высокого игрока позволяют ему иногда назначать и относительно высокую цену. Вероятность более высокой цены выше вероятности низкой.

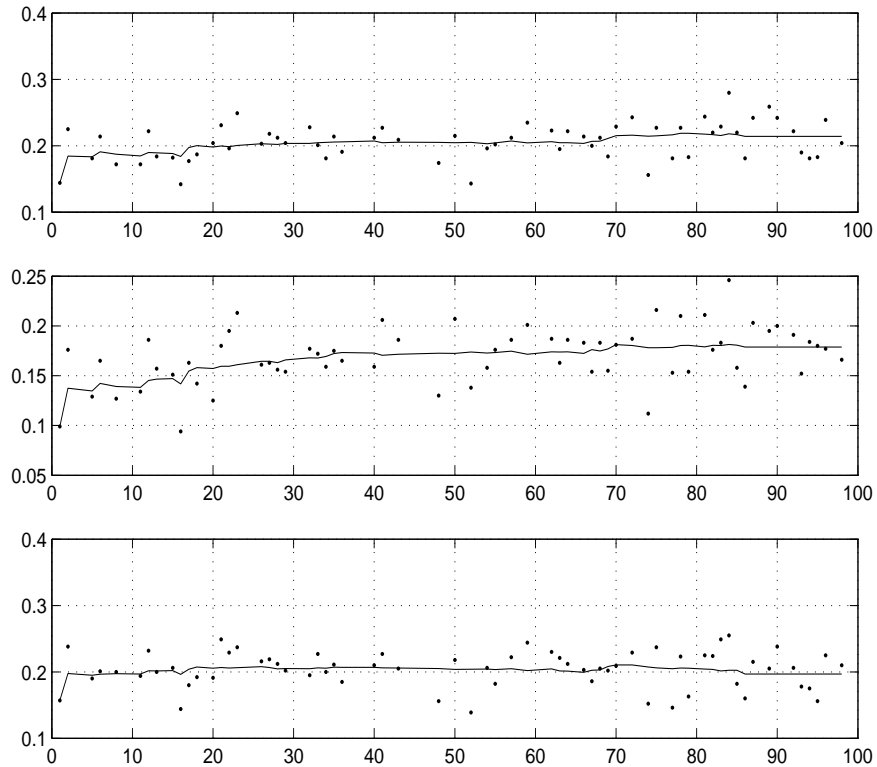


Рис. 3. Динамика индексов доходов (в расчете на один проект) трех игроков в игровых сериях, по 100 шагов каждая

Результаты являются до определенной степени неожиданными — первый игрок, не являясь безусловным лидером ни в одном из проектов, находит более успешные стратегии, чем каждый из лидеров. Третий игрок успешнее второго (при равенстве суммарных экспертных уровней). Поскольку вероятности проектов обоих типов одинаковы, то это означает, что переход уровня от 0.95 к 0.99 в одном из проектов компенсирует потери 0.75 : 0.71 в другом.

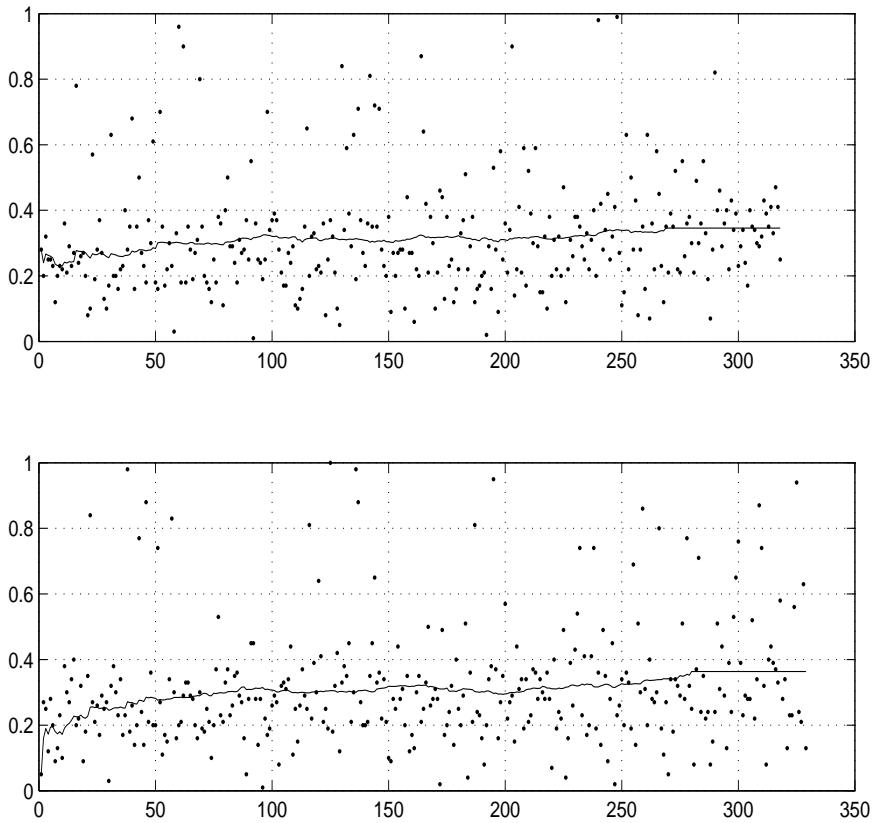


Рис. 4. Динамика аукционных ставок первого игрока
Показана каждая 10-я ставка и скользящее среднее по 50 ставкам. Видно, что после достаточно долгого равновесия в окрестности цены 0.31 первый игрок начал плавно повышать цены до значений 0.35, одновременно увеличив дисперсию своей смешанной стратегии, что привело к победе над 3-м игроком.

Таблица 2. Вероятности победы в аукционе для различных игроков

	Проект 1 [1,0]	Проект 2 [0,1]
Игрок 1	0.24	0.25
Игрок 2	0.44	0.46
Игрок 3	0.32	0.29

различаются.

Важный методологический момент — алгоритмы обучения с подкреплением, обобщенные на случай стохастических многошаговых игр, оказываются достаточно эффективными, сходимость достигается за несколько тысяч шагов одиночных игр, что является типичным значением для методик онлайн-обучения.

С прикладной точки зрения, имитационная модель игры COGITO позволяет изучать важные вопросы, касающиеся оптимизации бизнеса:

- Как влияет на доходы управляющей компании изменение ставки ее прибыли?
- Как при этом перераспределяются доходы отдельных компаний?
- Выгодно ли расширять холдинг путем включения компаний с относительно невысоким экспертным уровнем? Каковы пороговые значения для этого уровня?
- Выгодно ли одной компании повышать экспертный уровень другой компании (путем передачи технологий, обучения специалистов и пр.)?
- Каким должен быть экспертный профиль новой компании, чтобы обеспечить максимальную полезность холдингу без дестабилизации его работы?
- Насколько выгодно иметь экспертный уровень 0.999 в одной из тематик?

Список приложений далеко не исчерпан этим перечнем.

Послесловие

Сегодня теория игр — это, в основном, экономическая теория. Когда говорят о ее (всего лишь) полувекковой истории, то под этим понимается возраст

математической теории, но не самой игровой проблематики. Еще Платон, обращаясь к Сократу, обсуждал проблему о рациональном поведении дозорного на посту перед предстоящим боем. Если дозорный уверен в победе своего войска, то полезность его *личного* участия в бою исчезающе мала, а так как он может пострадать, то рациональнее будет покинуть пост до боя. Если же его наблюдения свидетельствуют о том, что противник сильнее, то тогда тем более разумно сбежать, так как это ничего не изменит, но он не пострадает. Этой же рациональной логике должны следовать и все другие воины. Но тогда поражение неизбежно!

На протяжении всей истории возникали проблемы подобного рода, но лишь начиная с работ Дж. фон Неймана в 30–40-х годах эти вопросы облекли в продуктивную математическую форму.

Фактически, теория игр пионерских времен фон Неймана, в ее математическом содержании, представляла собой совокупность отдельных моделей, описывающих изученные классы игр. Наиболее фундаментальным ее аспектом является теория индивидуальной полезности.

Полезность достаточно просто и корректно определяется для пары альтернатив, в смысле предпочтения одной альтернативы *в сравнении* с другой, но для полной теории ситуация несколько сложнее. Имея только сравнительный порядок довольно непросто выбрать универсальные метрические шкалы, позволяющие сравнивать полезности разных игроков и, тем самым, перейти к математическому понятию объективной полезности.

В теории игр любая информация об игре может иметь значение. Простой пример — должны ли игроки делать ходы в раундах повторяющейся игры одновременно или по очереди? Анализ равновесий Нэша в стратегических играх типа «дилеммы заключенного» показывает, что порядок ходов не влияет на равновесие. Однако вообразите себе (многошаговую игру) шахматы, в которых игроки делают ходы *одновременно* (сообщая их в конвертах судьбе; при невозможных парах ходов они игнорируются; трехкратный повтор невозможных комбинаций — ничья). Стратегии в ней *кардинально* будут отличаться от обычных шахмат.

Тотальная индуктивная рациональность игроков может приводить к логическим парадоксам, в которых следствия из рационального поведения приводят к заключению об иррациональности игрока (см. примеры в [20]). Для выхода из таких ситуаций требуется отказ от механически точного исполнения рациональных решений («дрогнувшая рука» Р. Зельтена — в ходах игроков должны допускаться ошибки).

В данной лекции обсуждались, в основном, механизмы машинного обу-

чения (machine learning) при выработке оптимальных стратегий. Другой взгляд на это проблему — эволюционная теория игр. Это направление также интенсивно развивается (см. [8] и цитированную там литературу).

В завершение нужно отметить, что практические применения теории игр широко опираются на вычислительное моделирование. Компьютерные модели, такие как разработанная автором система COGITO, способны учитывать особенности бизнес-процессов, прогнозировать и, что особенно ценно, оптимизировать деловую активность в практических условиях множества одновременно действующих участников.

Фактически система COGITO является автоматизированной торговой системой, поскольку вырабатываемые решения являются рекомендациями по ценам сделок в рыночных условиях. В контексте этой лекции рынок формируется всеми участниками, каждый из которых своими действиями влияет на складывающуюся цену. Однако используемый аукционный алгоритм может быть обобщен на случай, когда цены определяются небольшим числом крупных агентов, а остальные агенты, по-существу, выступают в роли трейдеров.

Благодарности

Автор выражает глубокую благодарность Ю. В. Тюменцеву, Н. Г. Макаренко и С. А. Шумскому за доброжелательную критику.

Литература

1. *Интрилигатор М.* Математические методы оптимизации и экономическая теория. — М.: Айрис-пресс, 2002. — 576 с.
2. *Хэрри М., Шредер Р.* 6 SIGMA. — М.: Эксмо, 2003. — 464 с.
3. *фон Нейман Дж., Моргенштерн О.* Теория игр и экономическое поведение. — М. Наука, 1970. — 707 с.
4. The Essential John Nash / Ed. by *Harold W. Kuhn* and *Sylvia Nasar*. Princeton Univ. Press, 2002. — 244 pp.
5. *Харишаньи Дж., Зельтен Р.* Общая теория выбора равновесия в играх. — СПб.: Экономическая школа, 2001. — 498 с.
6. *Айзекс Р.* Дифференциальные игры. — М.: Мир, 1967. — 479 с.

7. *Osborne M.J.* An introduction to game theory. (In press, Oxford University Press). Selected chapters from
URL: <http://www.chass.utoronto.ca/~osborne/igt/index.html>
8. *Hingston P., Kendall G.* Learning versus evolution in iterated prisoner's dilemma.
URL: <http://www.cs.nott.ac.uk/~gjk/papers/cec2004ph.pdf>
9. *Губко М. В., Новиков Д. А.* Теория игр в управлении организационными системами. – М.: Синтез, 2002. – 139 с.
10. Экономико-математическое моделирование / Под ред. *И. Н. Дрогобыцкого*. – М.: Экзамен, 2004. – 798 с.
11. *Брейман Л.* Задача о правилах останки // В кн.: «Прикладная комбинаторная математика». Сб. статей под ред. *Э. Беккенбаха*. – М.: Мир, 1968. – 362 с.
12. *Терехов С. А.* Нейродинамическое программирование автономных агентов // Лекция для Школы-семинара «Современные проблемы нейроинформатики». М.: МИФИ, 2004. – Часть 2. – с.111–139.
13. *Hu Junling, Wellman M.P.* Multiagent reinforcement learning: Theoretical framework and an algorithm.
URL: <http://ai.eecs.umich.edu/people/wellman/index.html>
14. *Tesauro G.* Extending Q-learning to general adaptive multi-agent systems // Advances in Neural Information Processing Systems **16** (NIPS'2003). MIT Press, Cambridge, MA, 2004.
URL: http://books.nips.cc/papers/files/nips16/NIPS2003_CN16.pdf
15. *Sutton R., Barto A.* Reinforcement learning: An introduction. – MIT Press, 1998.
16. URL: <http://plato.stanford.edu/entries/prisoner-dilemma/>
17. *Shoham Y., Powers R., Grenager T.* Multi-agent reinforcement learning: A critical survey. – Stanford Univ., 2003.
URL: http://robotics.stanford.edu/~shoham/www%20papers/MA_Learning_ACritical_Survey_2003_0516.pdf
18. *Mead W. C., Brown S. K., Jones R. D., Bowling P. S., Barnes C. W.* Optimization and control of a small-angle negative ion source using an on-line adaptive controller based on the connectionist normalized local spline neural network // Nuclear Instruments and Methods. – 1994. – Vol. A352, p. 309.
19. *Jones R. D., Lee Y. C., Qian S., Barnes C. W., Bisset K. R., Bruce G. M., Flake G. W., Lee K., Lee L. A., Mead W. C., O'Rourke M. K., Poli I., Thode L. E.* Nonlinear adaptive networks: A little theory, a few applications // In: *Proc. of the First Los Alamos Conference on Cognitive Modeling in System Control*, June 10–14, 1990, Santa Fe, NM.
20. *Ross D.* Game theory // In Stanford Encyclopedia of Philosophy.
URL: <http://plato.stanford.edu/entries/game-theory/>

Приложение. Нейросеть CNLS (Connectionist Normalized Local Splines)

Статистическая модель распределения значений многомерной случайной функции основывается на параметрическом представлении ее плотности. Моменты распределения являются вещественными многомерными функциями и для их эффективных аппроксимаций по конечной выборке применяются нейросетевые методы.

Модель локальных нейросетевых сплайнов основана на следующих построениях. Выберем систему нормируемых функций радиального базиса

$$\varphi(\vec{r}) = \frac{1}{\|\varphi\|} \varphi(\|\vec{r} - \vec{c}_k\|^2) \quad (\text{П1})$$

таких, что их сумма нигде в области аппроксимации не обращается в нуль, и каждая из функций локализована в окрестности своего максимума $\vec{r} = \vec{c}_k$. Простейший пример — функции Гаусса.

Для гладкой (почти везде) функции f запишем тождества:

$$f(\vec{r}) \equiv \frac{f(\vec{r}) \sum_k \varphi_k(\vec{r})}{\sum_j \varphi_j(\vec{r})} \equiv \frac{\sum_k f(\vec{r}) \cdot \varphi_k(\vec{r})}{\sum_j \varphi_j(\vec{r})}. \quad (\text{П2})$$

Далее перейдем к приближениям. Поскольку каждый член суммы в числителе (П2) представляет собой произведение гладкой функции f и локализованной базисной функции, то f может быть заменена ее отрезком ряда Тейлора. Если сохранить только члены вплоть до линейных, то получим представление

$$f(\vec{r}) \approx \frac{\sum_k (a_k + \vec{b}_k(\vec{r} - \vec{c}_k)) \cdot \varphi(\|\vec{r} - \vec{c}_k\|^2)}{\sum_k \varphi_k(\vec{r})}. \quad (\text{П3})$$

В этом соотношении формально можно считать значения набора коэффициентов $\{a_k, \vec{b}_k, \vec{c}_k\}$ произвольными. В итоге мы имеем дело с аппроксимационной моделью

$$f(\vec{r}) \approx \sum_k (a_k + \vec{b}_k(\vec{r} - \vec{c}_k)) \cdot \Phi_k(\vec{r}; \vec{c}_1 \dots \vec{c}_k), \quad (\text{П4})$$

которая по аналогии с моделями радиальных базисных функций названа авторами [18–19] нейросетевой. «Рецептивные поля» базисных нейронов вследствие процедуры нормализации существенно отличаются от исходных базисных функций φ , по которым проводилось разложение:

$$\Phi_k = \frac{\varphi(\|\vec{r} - \vec{c}_k\|^2)}{\sum_j \varphi(\|\vec{r} - \vec{c}_j\|^2)}. \quad (\text{П5})$$

В областях, где центры базисных функций распределены относительно разреженно, вблизи максимума φ_k лишь одна эта функция существенно отлична от нуля. В этом случае она доминирует в сумме в знаменателе и, следовательно, значение нейронной функции Φ_k близко к константе. В итоге, в области влияния этой функции общая аппроксимация, даваемая нейросетью, близка к линейной регрессии. Кусочно-линейные фрагменты автоматически гладко «сшиваются» при переходе к области соседней нейронной функции.

В областях высокой концентрации центров нейронная сеть CNLS фактически переходит в радиальную базисную сеть RBF, поскольку из-за высокой степени перекрытия сумма в знаменателе близка к константе.

При использовании полученной аппроксимации нужно учитывать, что нейронные функции Φ_k , располагающиеся на периферии области, занятой данными из моделируемой выборки, имеют пределы на бесконечности, равные 1.

Выбор значений параметров нейросетевой модели проводится в традиционной постановке обучения с учителем. Для обучающей выборки пар $\{\vec{r}_s, \vec{y}_s\}$ формулируется функция правдоподобия, которая максимизируется с использованием градиента по параметрам. Для гауссового распределения ошибок аппроксимации формулы для градиента получены в основном тексте лекции (18)–(20). Для аппроксимации математического ожидания и дисперсии используется пара соотношений (П4), при этом коэффициенты \vec{c}_k , отвечающие положениям центров нейронных функций, для простоты могут выбираться одинаковыми. В этом случае каждая из аппроксимируемых функций определяется своей парой (a, b) . Приведем здесь явный вид

производных нейросети:

$$\begin{aligned}\frac{\partial net(\vec{r})}{\partial a_k} &= \frac{\varphi(\|\vec{r} - \vec{c}_k\|^2)}{\sum_j \varphi(\|\vec{r} - \vec{c}_j\|^2)}, \\ \frac{\partial net(\vec{r})}{\partial b_{kn}} &= \frac{(r_n - c_{kn})\varphi(\|\vec{r} - \vec{c}_k\|^2)}{\sum_j \varphi(\|\vec{r} - \vec{c}_j\|^2)}, \\ \frac{\partial net(\vec{r})}{\partial c_{kn}} &= -\frac{b_{kn} \cdot \varphi(\|\vec{r} - \vec{c}_k\|^2)}{\sum_j \varphi(\|\vec{r} - \vec{c}_j\|^2)} + \\ &+ \frac{2(r_{kn} - c_{kn}) \cdot \varphi'(\|\vec{r} - \vec{c}_k\|^2)}{\sum_j \varphi(\|\vec{r} - \vec{c}_j\|^2)} \times \\ &\times [net(\vec{r}) - (a_k + \vec{b}_k \cdot (\vec{r} - \vec{c}_k))].\end{aligned}$$

В приложениях для ускорения вычислений выбор положений \vec{c}_k может проводиться внешним алгоритмом без использования этих производных. В частности, могут использоваться случайные выборки из обучающих данных, центры кластеров карты Кохонена, алгоритм k -means или другие алгоритмы. Положение центров может также диктоваться требованиями к точности аппроксимации.

Выбираемые при автоматическом обучении центры нейронов группируются в областях больших градиентов аппроксимируемой функции, что, конечно, является несомненным достоинством этой нейронной сети.

В завершение приведем пример (на рис. 5) аппроксимации функции моделью CNLS зашумленных данных для случая одной переменной. Обучение сети проводилось алгоритмом *RProp*.

Задачи

Задача 1. Ошибка в формуле

По статистике, в трети опубликованных в математических журналах статей со сложными выкладками впоследствии были найдены ошибки. Пусть читателю известно, что в формулах для обучения нейронной сети (18)–(20), (Пб) этой лекции с вероятностью 0.3 содержится ошибка. Читатель, который знакомится с текстом с целью:

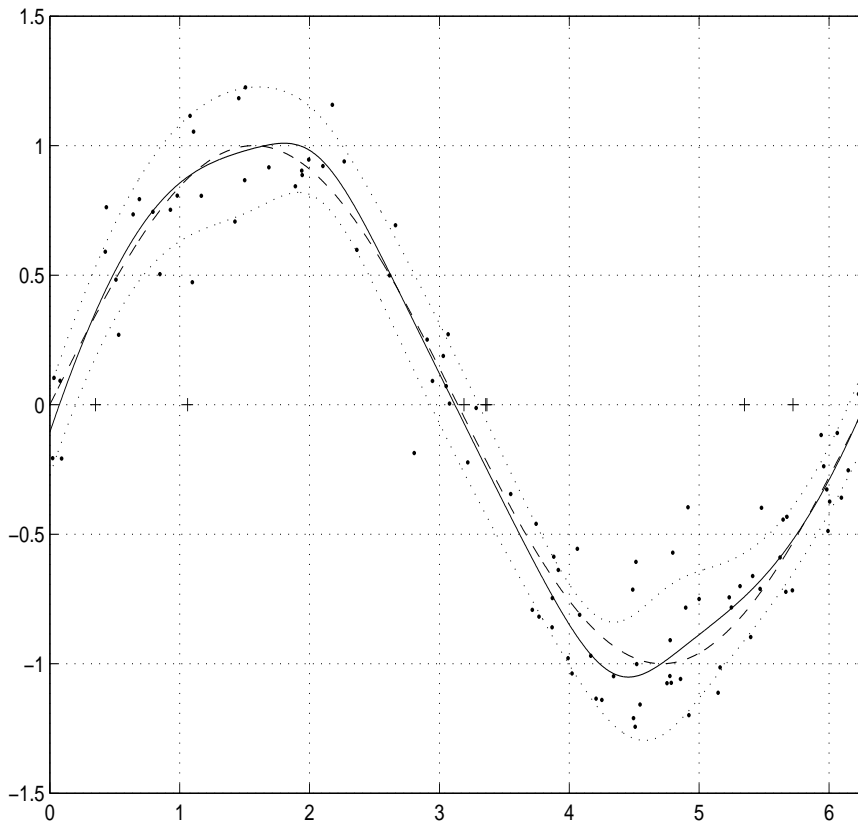


Рис. 5. Аппроксимация зашумленной синусоиды CNLS-сетью из семи нейронов
Пунктирные линии соответствуют прогнозируемым нейронной сетью значениям дисперсии. Крестиками (+) отмечены положения центров нейронов, выбранные алгоритмом обучения. Видно, что нейроны адаптивно располагаются в областях максимальных градиентов функции.

- **А** — общего любопытства,
- **Б** — выбора и дальнейшего совершенствования нейросетевых моделей,
- **В** — реализации нейросетевой модели в программе, которую предполагается продавать на рынке,

сталкивается с дилеммой — {*проверить, не проверять*} выкладки. Сформулируйте проблему выбора в форме стратегической игры двух игроков (читатель, автор) в матричной форме для случаев **А**, **Б** и **В**. Автор придерживается смешанной стратегии на своем множестве решений {*ошибка, нет ошибки*}, в которой вероятность ошибки $p = 0.3$. Какова оптимальная стратегия читателя в трех вариантах? Является ли эта стратегия чистой или смешанной? Как изменится решение, если стратегия автора неизвестна?

Задача 2. Плотность распределения континуума оптимальных решений в игре со стохастической матрицей

Алгоритм поиска «жадной» стратегии на шаге обучения с подкреплением (16)–(17) включает, как одну из возможностей, поиск плотности распределения значений аргументов, при которых стохастическая функция имеет максимум. Прямым путем для численной реализации такого поиска для случая конечного разбиения области определения (т. е. конечного дискретного набора возможных значений аргументов функции) является использование метода Монте-Карло.

Переход к более общему случаю континуума требует известной аккуратности при обращении с пределами. В этой связи предлагается следующая задача, являющаяся прелюдией к проблематике *бесконечных* игр.

Рассмотрим отрезок $[a, b]$, на котором задана пара непрерывных функций $m(x)$ и $\sigma(x)$, определяющих в каждой точке отрезка случайную величину с нормальным распределением $N(m(x), \sigma(x))$. Таким образом, на отрезке задана стохастическая функция (или случайный процесс, см. например, *Е. С. Венцель, Л. А. Овчаров. Теория случайных процессов и ее инженерные приложения. – М.: Наука, 1991*).

Разобьем исходный отрезок на множество отрезков длины ε , на каждом из них будем считать функции математического ожидания и дисперсии постоянными. Реализация случайной функции включает результат розыгрыша нормально распределенной случайной величины для каждого из отрезков ε -разбиения. Найдем максимум из полученного конечного набора чисел.

Пусть значение аргумента, при котором реализуется максимум в j -й реализации случайной функции равно x_j^* . Нас интересует плотность распределения величин x_j^* на отрезке $[a, b]$. Путем генерации новых реализаций исследуемой стохастической функции мы получаем выборочную оценку (гистограмму) $\rho_{j\varepsilon}(x)$ этой плотности, которая зависит от двух параметров — подробности разбиения ε и числа реализаций J .

Первый вопрос — корректен ли предельный переход

$$\rho_\varepsilon(x) \stackrel{?}{=} \lim_{J \rightarrow \infty} \rho(x).$$

Как будут устроены предельные функции (если они существуют)? Напомним, что нас интересует распределение значений аргументов максимумов, значения самих максимумов даже для нормального закона распределения в точке расходятся (убедитесь в этом).

Второй вопрос — при конечном числе реализаций J существует ли предел выборочной плотности при $\varepsilon \rightarrow 0$?

В какой метрике следует изучать эти пределы? Что можно сказать о задаче одновременного перехода к обоим пределам $J \rightarrow \infty$, $\varepsilon \rightarrow 0$?

Сергей Александрович ТЕРЕХОВ, кандидат физико-математических наук, начальник отдела интеллектуальных систем ООО «Нейрок Техсофт» (г. Троицк, Московская обл.). Область научных интересов — анализ данных при помощи искусственных нейронных сетей, генетические алгоритмы, марковские модели, байесовы сети, методы оптимизации, моделирование сложных систем. Автор 1 монографии и более 50 научных публикаций.

Л. РУТКОВСКИЙ

Кафедра вычислительной техники
Ченстоховского технического университета,
Ченстохова, Польша
E-mail: lrutko@kik.pcz.czyst.pl

НЕЙРО-НЕЧЕТКИЕ СИСТЕМЫ ВЫВОДА

Аннотация

В Лекции рассматривается новый класс нейро-нечетких систем вывода, основанных на логическом подходе и на подходе, предложенном Мамдани. Рассматриваются средства обеспечения гибкости при проектировании нейро-нечетких систем. Для предлагаемого подхода характерно автоматическое формирование правил нечеткого вывода в процессе обучения системы. Высокая точность нейро-нечетких систем демонстрируется на примере задачи моделирования динамического объекта.

LESZEK RUTKOWSKI

Department of Computer Engineering,
Technical University of Częstochowa,
Częstochowa, Poland
E-mail: lrutko@kik.pcz.czyst.pl

NEURO-FUZZY INFERENCE SYSTEMS

Abstract

In this paper we present a new class of neuro-fuzzy inference systems based on Mamdani and logical approaches. We introduce several flexibility concepts in the design of neuro-fuzzy systems. Our approach is characterized by automatic determination of fuzzy inference in the process of learning. A high accuracy of neuro-fuzzy systems is demonstrated in simulations of a dynamic plant.

Introduction

In the literature various neuro-fuzzy inference systems (NFIS) have been proposed (see e.g. [1–5], [16–20]). They have been used in a wide range of problems, e.g. system identification, pattern classification, system control, prediction and image processing. It is well known that traditional fuzzy systems suffer from the lack of learning properties. On the other hand neural networks are not able to incorporate a linguistic information coming from human experts. Neuro-fuzzy systems presented by several authors (see e.g. [7–12]) exhibit advantages of neural networks and fuzzy systems, i.e. they combine the natural language description of fuzzy systems and the learning properties of neural networks. In this paper we develop a new class of neuro-fuzzy systems. Our approach is characterized by automatic determination of fuzzy inference in the process of learning. It is well known that introducing additional parameters to be tuned in neuro-fuzzy systems improves their performance and they are able to better represent the patterns encoding in data. Therefore, in this paper we introduce several flexibility concepts in the design of neuro-fuzzy systems. Due to additional parameters incorporated into a neuro-fuzzy system, we achieve an excellent performance of neuro-fuzzy systems. A high accuracy of neuro-fuzzy systems is demonstrated in simulations of a dynamic plant. In the paper by T , S , and N we denote t -norm, t -conorm and negation, respectively.

Non-flexible neuro-fuzzy systems

Description of fuzzy systems

In this paper, we consider multi-input-single-output fuzzy system mapping $X \rightarrow Y$, where $X \subset R^n$ and $Y \subset R$. The system (see Fig. 1) is composed of a fuzzifier, a fuzzy rule base, a fuzzy inference engine and a defuzzifier. The fuzzifier performs a mapping from the observed crisp input to a fuzzy set. The most commonly used fuzzifier is the singleton fuzzifier which maps $\bar{\mathbf{x}} = [\bar{x}_1, \dots, \bar{x}_n] \in X$ into a fuzzy set $A' \subseteq X$ characterized by the membership function

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \bar{\mathbf{x}} \\ 0 & \text{if } \mathbf{x} \neq \bar{\mathbf{x}} \end{cases} . \quad (1)$$

The fuzzy rule base consists of a collection of N fuzzy IF-THEN rules,

aggregated by the disjunction or the conjunction, in the form

$$R^{(k)} : \begin{cases} \text{IF } x_1 \text{ is } A_1^k \text{ AND} \\ \quad x_2 \text{ is } A_2^k \text{ AND } \dots \\ \quad x_n \text{ is } A_n^k \\ \text{THEN } y \text{ is } B^k \end{cases} \quad (2)$$

or

$$R^{(k)} : \text{IF } \mathbf{x} \text{ is } A^k \text{ THEN } y \text{ is } B^k, \quad (3)$$

where $\mathbf{x} = [x_1, \dots, x_n] \in X$, $y \in Y$, $A^k = A_1^k \times A_2^k \times \dots \times A_n^k$, $A_1^k, A_2^k, \dots, A_n^k$ are fuzzy sets characterized by membership functions

$$\mu_{A_i^k}(x_i), \quad i = 1, \dots, n; \quad k = 1, \dots, N,$$

whereas B^k are fuzzy sets characterized by membership functions $\mu_{B^k}(y)$, $k = 1, \dots, N$. The firing strength of the k -th rule, $k = 1, \dots, N$, is defined by

$$\tau_k(\bar{\mathbf{x}}) = \prod_{i=1}^n \{\mu_{A_i^k}(\bar{x}_i)\} = \mu_{A^k}(\bar{\mathbf{x}}). \quad (4)$$

In the paper notations τ_k and $\mu_{A^k}(\bar{\mathbf{x}})$ will be used interchangeably.

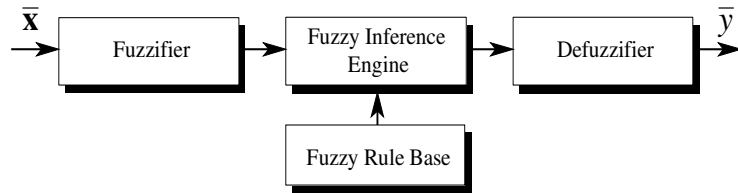


Figure 1. Fuzzy inference system

The fuzzy inference engine determines the mapping from the fuzzy sets in the input space X to the fuzzy sets in the output space Y . Each of N rules (3) determines a fuzzy set $\bar{B}^k \subseteq Y$ given by the compositional rule of inference

$$\bar{B}^k = A' \circ (A^k \rightarrow B^k), \quad (5)$$

where $A^k = A_1^k \times A_2^k \times \dots \times A_n^k$. Fuzzy sets \bar{B}^k are characterized by membership functions expressed by the sup-star composition

$$\mu_{\bar{B}^k}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ \mu_{A'}(\mathbf{x}) * \mu_{A_1^k \times \dots \times A_n^k \rightarrow B^k}(\mathbf{x}, y) \right\}, \quad (6)$$

where $*$ can be any operator in the class of t -norms. It is easily seen that for a crisp input $\bar{\mathbf{x}} = [\bar{x}_1, \dots, \bar{x}_n] \in \mathbf{X}$, i.e., the singleton fuzzifier (1), formula (6) becomes

$$\begin{aligned} \mu_{\bar{B}^k}(y) &= \mu_{A_1^k \times \dots \times A_n^k \rightarrow B^k}(\bar{\mathbf{x}}, y) \\ &= \mu_{A^k \rightarrow B^k}(\bar{\mathbf{x}}, y) \\ &= I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)), \end{aligned} \quad (7)$$

where $I(\cdot)$ is given by

$$I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)) = T\{\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)\}. \quad (8)$$

in the case of the Mamdani approach or by

$$I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)) = S\{1 - \mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)\}. \quad (9)$$

in the case of the logical approach. In equation (9) we use an S -implication. We refer to (8) and (9) as to “engineering implication” or “fuzzy implication”, respectively (see [16]). Typical examples of the S -implication include the following formulas:

a) The Kleene-Dienes implication

$$I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)) = \max\{1 - \mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)\} \quad (10)$$

b) The Lukasiewicz implication

$$I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)) = \min\{1, 1 - \mu_{A^k}(\bar{\mathbf{x}}) + \mu_{B^k}(y)\} \quad (11)$$

c) The Reichenbach implication

$$I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)) = 1 - \mu_{A^k}(\bar{\mathbf{x}}) + \mu_{A^k}(\bar{\mathbf{x}}) \cdot \mu_{B^k}(y) \quad (12)$$

d) The Fodor implication

$$I(\mu_{A^k}(\bar{x}), \mu_{B^k}(y)) = \begin{cases} 1 & \text{if } \mu_{A^k}(\bar{x}) \leq \mu_{B^k}(y) \\ \max\{1 - \mu_{A^k}(\bar{x}), \mu_{B^k}(y)\} & \text{if } \mu_{A^k}(\bar{x}) > \mu_{B^k}(y) \end{cases} \quad (13)$$

Obviously, an S -implication can be generated by other t -conorms. The aggregation operator, applied in order to obtain the fuzzy set B' based on fuzzy sets \bar{B}^k , is the t -norm or t -conorm operator, depending on the type of fuzzy implication. In case of the Mamdani approach, the aggregation is carried out by

$$B' = \bigcup_{k=1}^N \bar{B}^k. \quad (14)$$

The membership function of B' is computed by the use of a t -conorm, that is

$$\mu_{B'}(y) = \bigvee_{k=1}^N \mu_{\bar{B}^k}(y). \quad (15)$$

When we use the logical model, the aggregation is carried out by

$$\bar{B} = \bigcap_{k=1}^N \bar{B}^k. \quad (16)$$

The membership function of B' is determined by the use of a t -norm, i.e.

$$\mu_{B'}(y) = \bigwedge_{k=1}^N \{\mu_{\bar{B}^k}(y)\}. \quad (17)$$

The defuzzifier performs a mapping from the fuzzy set B' to a crisp point \bar{y} in $Y \subset R$. The COA (center of area) method is defined by the following formula

$$\bar{y} = \frac{\int_Y y \cdot \mu_{B'}(y) dy}{\int_Y \mu_{B'}(y) dy} \quad (18)$$

or by

$$\bar{y} = \frac{\sum_{r=1}^N \bar{y}^r \cdot \mu_{B'}(\bar{y}^r)}{\sum_{r=1}^N \mu_{B'}(\bar{y}^r)} \quad (19)$$

in the discrete form, where \bar{y}^r are centers of the membership functions $\mu_{B^r}(y)$, i.e., for $r = 1, \dots, N$

$$\mu_{B^r}(\bar{y}^r) = \max_{y \in Y} \{\mu_{B^r}(y)\}. \quad (20)$$

Mamdani-type neuro-fuzzy systems

In this approach function $I(\cdot)$ is given by formula (8). Consequently,

$$I(\mu_{A^k}(\bar{x}), \mu_{B^k}(\bar{y}^r)) = T\{\mu_{A^k}(\bar{x}), \mu_{B^k}(\bar{y}^r)\}. \quad (21)$$

The aggregated output fuzzy set $B' \subseteq Y$ is given by

$$\mu_{B'}(\bar{y}^r) = \sum_{k=1}^N \mu_{\bar{B}^k}(\bar{y}^r) = \sum_{k=1}^N \{T\{\mu_{A^k}(\bar{x}), \mu_{B^k}(\bar{y}^r)\}\}. \quad (22)$$

Consequently, formula (19) takes the form

$$\bar{y} = \frac{\sum_{r=1}^N \bar{y}^r \cdot \sum_{k=1}^N \left\{ T \left\{ \prod_{i=1}^n \mu_{A_i^k}(\bar{x}_i), \mu_{B^k}(\bar{y}^r) \right\} \right\}}{\sum_{r=1}^N \sum_{k=1}^N \left\{ T \left\{ \prod_{i=1}^n \mu_{A_i^k}(\bar{x}_i), \mu_{B^k}(\bar{y}^r) \right\} \right\}}. \quad (23)$$

Logical-type neuro-fuzzy systems

In this approach function $I(\cdot)$ is given by formula (9). Therefore

$$I(\mu_{A^k}(\bar{x}), \mu_{B^k}(\bar{y}^r)) = S\{1 - \mu_{A^k}(\bar{x}), \mu_{B^k}(\bar{y}^r)\}. \quad (24)$$

The aggregated output fuzzy set $B' \subseteq Y$ is given by

$$\begin{aligned} \mu_{B'}(\bar{y}^r) &= \sum_{k=1}^N \mu_{\bar{B}^k}(\bar{y}^r) \\ &= \sum_{k=1}^N \{S\{N(\mu_{A^k}(\bar{x})), \mu_{B^k}(\bar{y}^r)\}\}. \end{aligned} \quad (25)$$

and formula (19) becomes

$$\bar{y} = \frac{\sum_{r=1}^N \bar{y}^r \cdot \sum_{k=1}^N \left\{ S \left\{ N \left(\prod_{i=1}^n \mu_{A_i^k}(\bar{x}_i) \right), \mu_{B^k}(\bar{y}^r) \right\} \right\}}{\sum_{r=1}^N \sum_{k=1}^N \left\{ S \left\{ N \left(\prod_{i=1}^n \mu_{A_i^k}(\bar{x}_i) \right), \mu_{B^k}(\bar{y}^r) \right\} \right\}}. \quad (26)$$

Generalized neuro-fuzzy systems

We will generalize both approaches described in sections “Mamdani-type neuro-fuzzy systems” and “Logical-type neuro-fuzzy systems” and propose a general architecture of fuzzy systems. Observe that systems (23) and (26) can be presented in the form

$$\bar{y} = f(\bar{x}) = \frac{\sum_{r=1}^N \bar{y}^r \cdot \text{agr}_r(\bar{x}, \bar{y}^r)}{\sum_{r=1}^N \text{agr}_r(\bar{x}, \bar{y}^r)}, \quad (27)$$

where

$$\text{agr}_r(\bar{x}, \bar{y}^r) = \begin{cases} S_{k=1}^N \{I_{k,r}(\bar{x}, \bar{y}^r)\} & \text{for the Mamdani approach} \\ T_{k=1}^N \{I_{k,r}(\bar{x}, \bar{y}^r)\} & \text{for the logical approach} \end{cases} \quad (28)$$

$$I_{k,r}(\bar{x}, \bar{y}^r) = \begin{cases} T\{\tau_k(\bar{x}), \mu_{B^k}(\bar{y}^r)\} & \text{for the Mamdani approach} \\ S(1 - \tau_k(\bar{x}), \mu_{B^k}(\bar{y}^r)) & \text{for the logical approach} \end{cases}. \quad (29)$$

The firing strength of rules τ_k is defined by formula (4). The general architecture of system (27) is depicted in Fig. 2.

Flexibility in fuzzy systems

We will present various concepts leading to the designing flexible neuro-fuzzy systems.

Compromise operators

In this section we propose the following combination of the Mamdani-type and logical-type systems:

$$I(\mu_{A^k}(\bar{x}), \mu_{B^k}(y)) = (1 - \lambda) T\{\mu_{A^k}(\bar{x}), \mu_{B^k}(y)\} + \lambda S\{1 - \mu_{A^k}(\bar{x}), \mu_{B^k}(y)\} \quad (30)$$

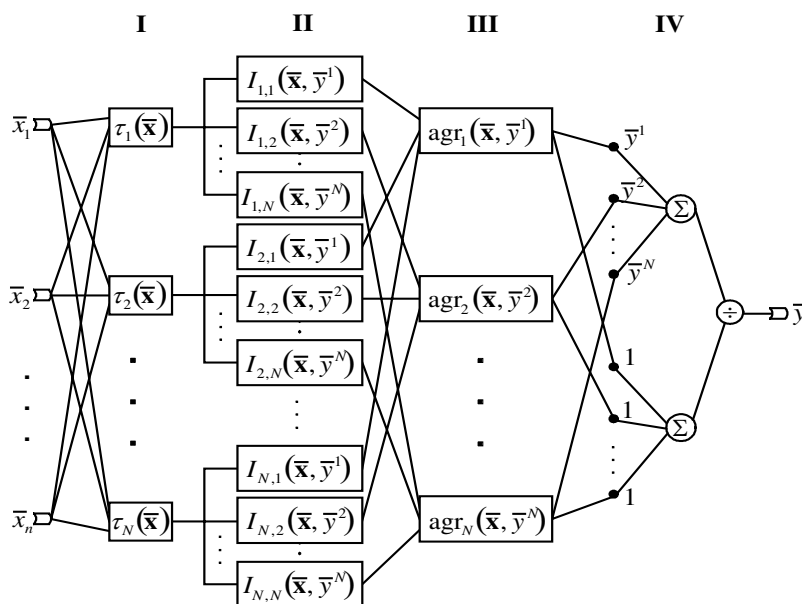


Figure 2. General architecture of fuzzy systems studied in the paper (flexible and nonflexible)

where parameter λ should satisfy $0 \leq \lambda \leq 1$.

Example 1. As an example we define the following combination

$$I(\mu_{A^k}(\bar{x}), \mu_{B^k}(y)) = (1 - \lambda) \min \{ \mu_{A^k}(\bar{x}), \mu_{B^k}(y) \} + \lambda \max \{ 1 - \mu_{A^k}(\bar{x}), \mu_{B^k}(y) \} \quad (31)$$

Weighted triangular norms

We propose the weighted t -norm

$$T^* \{ a_1, \dots, a_n; w_1^T, \dots, w_n^T \} = \frac{n}{T} \{ 1 - w_i^T (1 - \alpha_i) \} \quad (32)$$

to connect the antecedents in each rule, $k = 1, \dots, N$, and the weighted t -norm and t -conorm:

$$T^* \{a_1, \dots, a_N; w_1^{\text{agr}}, \dots, w_N^{\text{agr}}\} = \prod_{k=1}^N \{1 - w_k^{\text{agr}}(1 - a_k)\} \quad (33)$$

$$S^* \{a_1, \dots, a_N; w_1^{\text{agr}}, \dots, w_N^{\text{agr}}\} = \prod_{k=1}^N \{w_k^{\text{agr}} a_k\} \quad (34)$$

to aggregate the individual rules in the logical and Mamdani models, respectively. The weights w_i^t , w_k and w_k^{agr} are in the interval $[0,1]$. It is easily seen that formula (32) can be applied to the evaluation of an importance of input linguistic values, and the weighted t -norm (33) or t -conorm (34) to a selection of important rules.

Parameterized triangular norms

Parameterized triangular norms include the Dombi, Hamacher, Yager, Frank, Weber I, Weber II, Dubois-Prade and other families [6]. We use notation $\overleftrightarrow{T} \{a_1, a_2, \dots, a_n; p\}$ and $\overleftrightarrow{S} \{a_1, a_2, \dots, a_n; p\}$ for parameterized triangular norms. The hyperplanes corresponding to them can be adjusted in the process of learning of parameter p . As an example we present the Dombi family of parameterized triangular norms. The t -norm and t -conorm are given as follows:

a) The Dombi t -norm

$$\overleftrightarrow{T} \{\mathbf{a}; p\} = \begin{cases} \text{drastic } t\text{-norm} & \text{for } p = 0 \\ \frac{1}{1 + \left(\sum_{i=1}^n \left(\frac{1 - a_i}{a_i} \right)^p \right)^{\frac{1}{p}}} & \text{for } p \in (0, \infty) \\ \text{Zadeh } t\text{-norm} & \text{for } p = \infty \end{cases} \quad (35)$$

where \overleftrightarrow{T} stands for a t -norm of the Dombi family parameterized by p .

b) The Dombi t -conorm

$$\overleftrightarrow{S}\{\mathbf{a}; p\} = \begin{cases} \text{drastic } t\text{-conorm} & \text{for } p = 0 \\ 1 - \frac{1}{1 + \left(\sum_{i=1}^n \left(\frac{a_i}{1-a_i}\right)^p\right)^{\frac{1}{p}}} & \text{for } p \in (0, \infty) \\ \text{Zadeh } t\text{-conorm} & \text{for } p = \infty \end{cases} \quad (36)$$

where \overleftrightarrow{S} stands for a t -conorm of the Dombi family parameterized by p .

Combining the S -implication and (36) we get the fuzzy S -implication generated by the Dombi family

$$\overleftrightarrow{I}(a, b; p) = 1 - \frac{1}{1 + \left(\left(\frac{1-a}{a}\right)^p + \left(\frac{b}{1-b}\right)^p\right)^{\frac{1}{p}}}. \quad (37)$$

Soft fuzzy norms

In this section we recall a concept of soft fuzzy norms proposed by Yager and Filev [21]. Let a_1, \dots, a_n be numbers in the unit interval that are to be aggregated. The soft version of triangular norms suggested by Yager and Filev is defined by

$$\tilde{T}\{\mathbf{a}; \alpha\} = (1 - \alpha) \frac{1}{n} \sum_{i=1}^n a_i + \alpha \overleftrightarrow{T}_{i=1}^n \{a_i\} \quad (38)$$

$$\tilde{S}\{\mathbf{a}; \alpha\} = (1 - \alpha) \frac{1}{n} \sum_{i=1}^n a_i + \alpha \overleftrightarrow{S}_{i=1}^n \{a_i\} \quad (39)$$

where $\alpha \in [0, 1]$. They allow to balance between the arithmetic average aggregator and the triangular norm aggregator depending on parameter α .

Example 2. (An example of soft algebraic triangular norms) The soft algebraic triangular norms are based on classical algebraic triangular norms (see e.g. [6]). The soft algebraic t -norm is described as follows

$$\tilde{T}\{a_1, a_2; \alpha\} = (1 - \alpha) \frac{1}{2} (a_1 + a_2) + \alpha a_1 a_2. \quad (40)$$

Flexible neuro-fuzzy systems

In this section we will derive flexible neuro-fuzzy systems by making use of the concepts presented in section “Flexibility in fuzzy systems”. We start with a definition which is a generalization of a strong negation (see [6]).

Definition 1. (Compromise operator) Function

$$\tilde{N}_\nu : [0, 1] \rightarrow [0, 1] \quad (41)$$

given by

$$\begin{aligned} \tilde{N}_\nu(a) &= (1 - \nu)N(a) + \nu N(N(a)) \\ &= (1 - \nu)N(a) + \nu a \end{aligned} \quad (42)$$

is called a compromise operator where $\nu \in [0, 1]$ and $N(a) = \tilde{N}_0(a) = 1 - a$.

Observe that

$$\tilde{N}_\nu(a) = \begin{cases} N(a) & \text{for } \nu = 0 \\ \frac{1}{2} & \text{for } \nu = \frac{1}{2} \\ a & \text{for } \nu = 1 \end{cases} \quad (43)$$

Obviously function \tilde{N}_ν is a strong negation for $\nu = 0$. The 3D plot of function (42) is depicted in Fig. 3.

Definition 3. (H -function) Function

$$H : [0, 1]^n \rightarrow [0, 1]$$

given by

$$\begin{aligned} H(\mathbf{a}; \nu) &= \tilde{N}_\nu \left(\overset{n}{S}_{i=1} \left\{ \tilde{N}_\nu(a_i) \right\} \right) \\ &= \tilde{N}_{1-\nu} \left(\overset{n}{T}_{i=1} \left\{ \tilde{N}_{1-\nu}(a_i) \right\} \right) \end{aligned} \quad (44)$$

is called an H -function where $\nu \in [0, 1]$.

Theorem 1. Let T and S be dual triangular norms. Function H defined by (44) varies between a t -norm and a t -conorm as ν goes from 0 to 1.

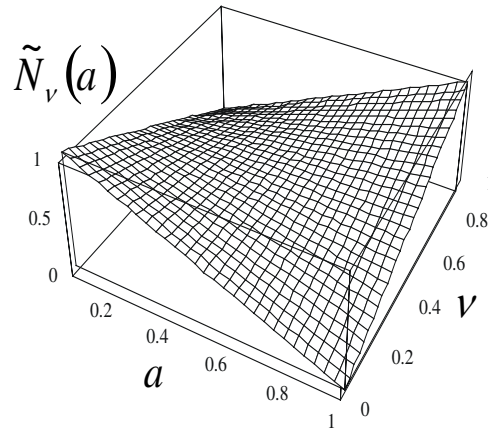


Figure 3. 3D plot of function (42)

Proof. From the assumption it follows that

$$T\{\mathbf{a}\} = N(S\{N(a_1), N(a_2), \dots, N(a_n)\}). \quad (45)$$

For $\nu = 0$ formula (45) can be rewritten with the notation of the compromise operator (42)

$$T\{\mathbf{a}\} = \tilde{N}_0(S\{\tilde{N}_0(a_1), \tilde{N}_0(a_2), \dots, \tilde{N}_0(a_n)\}). \quad (46)$$

Apparently

$$S\{\mathbf{a}\} = \tilde{N}_1(S\{\tilde{N}_1(a_1), \tilde{N}_1(a_2), \dots, \tilde{N}_1(a_n)\}) \quad (47)$$

for $\nu = 1$.

The right-hand sides of (46) and (47) can be written as follows

$$H(\mathbf{a}; \nu) = \tilde{N}_\nu\left(\tilde{S}_{i=1}^n\{\tilde{N}_\nu(a_i)\}\right) \quad (48)$$

for $\nu = 0$ and $\nu = 1$, respectively. If parameter ν changes from 0 to 1, then the result is established.

Remark 1. Observe that

$$H(\mathbf{a}; \nu) = \begin{cases} T\{\mathbf{a}\} & \text{for } \nu = 0 \\ \frac{1}{2} & \text{for } \nu = \frac{1}{2} \\ S\{\mathbf{a}\} & \text{for } \nu = 1. \end{cases} \quad (49)$$

It is easily seen, that for $0 < \nu < 0.5$ the H -function resembles a t -norm and for $0.5 < \nu < 1$ the H -function resembles a t -conorm.

Example 3. (An example of the H -function generated by the algebraic triangular norms) We will apply Theorem 1 to illustrate (for $n = 2$) how to switch between the algebraic t -norm

$$T\{a_1, a_2\} = H(a_1, a_2; 0) = a_1 a_2 \quad (50)$$

and the algebraic t -conorm

$$S\{a_1, a_2\} = H(a_1, a_2; 1) = a_1 + a_2 - a_1 a_2. \quad (51)$$

The H -function generated by formulas (50) or (51) takes the form

$$\begin{aligned} H(a_1, a_2; \nu) &= \tilde{N}_{1-\nu}(\tilde{N}_{1-\nu}(a_1) \tilde{N}_{1-\nu}(a_2)) \\ &= \tilde{N}_\nu\left(1 - \left(1 - \tilde{N}_\nu(a_1)\right) \left(1 - \tilde{N}_\nu(a_2)\right)\right) \end{aligned} \quad (52)$$

and varies from (50) to (51) as ν goes from zero to one. In Fig. 4, we illustrate function (52) for $\nu = 0.00, \nu = 0.15, \nu = 0.50, \nu = 0.85, \nu = 1.00$.

Theorem 2. Let T and S be dual triangular norms.

Then

$$I(a, b; \nu) = H(\tilde{N}_{1-\nu}(a), b; \nu) \quad (53)$$

switches between an “engineering implication”

$$I_{\text{eng}}(a, b) = I(a, b; 0) = T\{a, b\} \quad (54)$$

and an S -implication

$$I_{\text{fuzzy}}(a, b) = I(a, b; 1) = S\{1 - a, b\} \quad (55)$$

Proof. Theorem 2 is a straightforward consequence of Theorem 1.

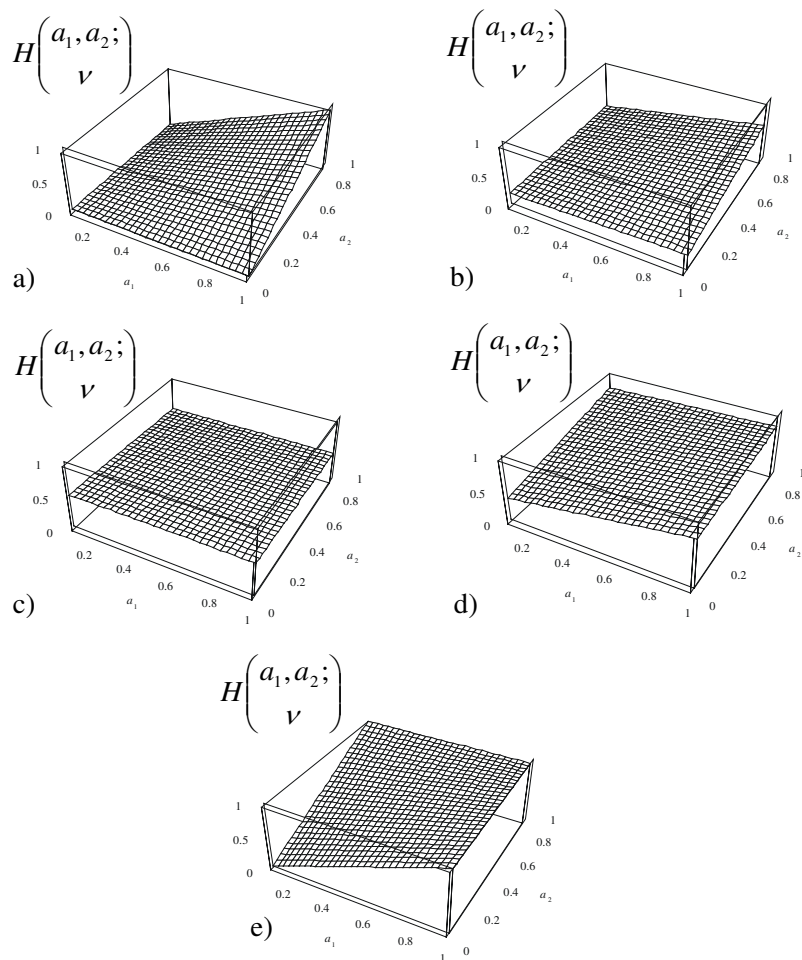


Figure 4. 3D plots of function (52) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$.

Example 4. (An example of the H -implication generated by the algebraic triangular norms) We will define the H -implication generated by the algebraic triangular norms. Let

$$\begin{aligned} I_{\text{eng}}(a, b) &= H(a, b; 0) \\ &= T\{a, b\} \\ &= ab \end{aligned} \quad (56)$$

$$\begin{aligned} I_{\text{fuzzy}}(a, b) &= H(\tilde{N}_0(a), b; 1) \\ &= S\{N(a), b\} \\ &= 1 - a + ab. \end{aligned} \quad (57)$$

Then

$$I(a, b; \nu) = H(\tilde{N}_{1-\nu}(a), b; \nu) \quad (58)$$

goes from (56) to (57) as ν varies from 0 to 1. The 3D plots of function (58) are depicted in Fig. 5.

The concept of generalized triangular norms leads to the following flexible neuro-fuzzy systems:

$$\bar{y} = f(\bar{x}) = \frac{\sum_{r=1}^N \bar{y}^r \cdot \text{agr}_r(\bar{x}, \bar{y}^r)}{\sum_{r=1}^N \text{agr}_r(\bar{x}, \bar{y}^r)} \quad (59)$$

where

$$\tau_k(\bar{x}) = \left(\begin{array}{l} (1 - \alpha^\tau) \text{avg}(\mu_{A_1^k}(\bar{x}_1), \dots, \mu_{A_n^k}(\bar{x}_n)) + \\ + \alpha^\tau \overset{\leftrightarrow}{H} \left(\mu_{A_1^k}(\bar{x}_1), \dots, \mu_{A_n^k}(\bar{x}_n); \right. \\ \left. w_{1,k}^\tau, \dots, w_{n,k}^\tau, p^\tau, 0 \right) \end{array} \right) \quad (60)$$

$$I_{k,r}(\bar{x}, \bar{y}^r) = \left(\begin{array}{l} (1 - \alpha^I) \text{avg}(\tilde{N}_{1-\nu}(\tau_k(\bar{x})), \mu_{B^k}(\bar{y}^r)) + \\ + \alpha^I \overset{\leftrightarrow}{H} \left(\tilde{N}_{1-\nu}(\tau_k(\bar{x})), \mu_{B^k}(\bar{y}^r); \right. \\ \left. p^I, \nu \right) \end{array} \right) \quad (61)$$

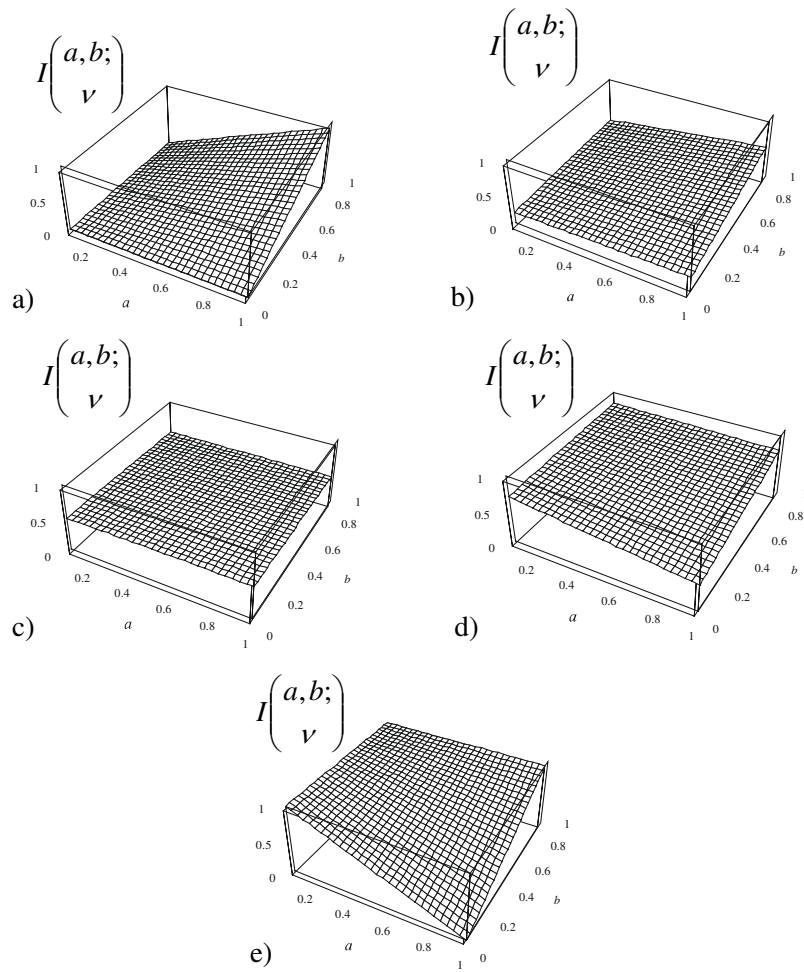


Figure 5. 3D plots of function (58) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$.

$$\text{agr}_r(\bar{x}, \bar{y}^r) = \left(\begin{array}{l} (1 - \alpha^{\text{agr}}) \text{avg}(I_{1,r}(\bar{x}, \bar{y}^r), \dots, I_{N,r}(\bar{x}, \bar{y}^r)) + \\ + \alpha^{\text{agr}} \overleftrightarrow{H}^* \left(\begin{array}{l} I_{1,r}(\bar{x}, \bar{y}^r), \dots, I_{N,r}(\bar{x}, \bar{y}^r); \\ w_1^{\text{agr}}, \dots, w_N^{\text{agr}}, p^{\text{agr}}, 1 - \nu \end{array} \right) \end{array} \right) \quad (62)$$

In the above system we use parameterised families $\overleftrightarrow{H}(\cdot)$ and parameterised families with weights $\overleftrightarrow{H}^*(\cdot)$ analogously to formula (35) and (33). More specifically, in (60) and (62) we use the following definition

$$\overleftrightarrow{H}^* \left(\begin{array}{l} a_1, \dots, a_n; \\ w_1, \dots, w_n, p, \nu \end{array} \right) = \overleftrightarrow{H} \left(\begin{array}{l} \text{arg}_1(a_1, w_1, \nu), \dots, \text{arg}_n(a_n, w_n, \nu); \\ p, \nu \end{array} \right) \quad (63)$$

where

$$\text{arg}_i(a_i, w_i, \nu) = (1 - \nu)(1 - w_i(1 - a_i)) + \nu w_i a_i. \quad (64)$$

Compromise neuro-fuzzy systems

In this section we will derive neuro-fuzzy systems based on compromise operators presented in section “Compromise operators”. The compromise neuro-fuzzy systems are given by the following formulas:

$$\bar{y} = f(\bar{x}) = \frac{\sum_{r=1}^N \bar{y}^r \cdot \text{agr}_r(\bar{x}, \bar{y}^r)}{\sum_{r=1}^N \text{agr}_r(\bar{x}, \bar{y}^r)} \quad (65)$$

where

$$\tau_k(\bar{x}) = \left(\begin{array}{l} (1 - \alpha^\tau) \text{avg}(\mu_{A_1^k}(\bar{x}_1), \dots, \mu_{A_n^k}(\bar{x}_n)) + \\ + \alpha^\tau \overleftrightarrow{T}^* \left(\begin{array}{l} \mu_{A_1^k}(\bar{x}_1), \dots, \mu_{A_n^k}(\bar{x}_n); \\ w_{1,k}^\tau, \dots, w_{n,k}^\tau, p^\tau \end{array} \right) \end{array} \right) \quad (66)$$

$$I_{k,r}(\bar{x}, \bar{y}^r) = \left(\begin{array}{l} (1 - \alpha^I) \text{avg}(\tilde{N}_{1-\lambda}(\tau_k(\bar{x}), \mu_{B^k}(\bar{y}^r)) + \\ + \alpha^I \left(\begin{array}{l} (1 - \lambda) \overleftrightarrow{T} \left(\begin{array}{l} \tau_k(\bar{x}), \mu_{B^k}(\bar{y}^r); \\ p^I \end{array} \right) + \\ + \lambda \overleftrightarrow{S} \left(\begin{array}{l} 1 - \tau_k(\bar{x}), \mu_{B^k}(\bar{y}^r); \\ p^I, \nu \end{array} \right) \end{array} \right) \end{array} \right) \quad (67)$$

$$\begin{aligned}
 agr_r(\bar{\mathbf{x}}, \bar{y}^r) &= \\
 &= \left(\begin{array}{l} (1 - \alpha^{\text{agr}}) \text{avg} (I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \dots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r)) + \\ + \alpha^{\text{agr}} \left(\begin{array}{l} (1 - \lambda) \overset{\leftrightarrow}{S} \left\{ I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \dots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); \right\} + \\ \lambda \overset{\leftrightarrow}{T} \left\{ I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \dots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); \right\} \\ w_1^{\text{agr}}, \dots, w_N^{\text{agr}}, p^{\text{agr}} \end{array} \right) \end{array} \right) \quad (68)
 \end{aligned}$$

The above three formulas can be rewritten by making use of the H -function concept:

$$\tau_k(\bar{\mathbf{x}}) = \left(\begin{array}{l} (1 - \alpha^\tau) \text{avg} \left(\mu_{A_1^k}(\bar{x}_1), \dots, \mu_{A_n^k}(\bar{x}_n) \right) + \\ + \alpha^\tau \overset{\leftrightarrow}{H} \left(\begin{array}{l} \mu_{A_1^k}(\bar{x}_1), \dots, \mu_{A_n^k}(\bar{x}_n); \\ w_{1,k}^\tau, \dots, w_{n,k}^\tau, p^\tau, 0 \end{array} \right) \end{array} \right) \quad (69)$$

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = \left(\begin{array}{l} (1 - \alpha^I) \text{avg} \left(\tilde{N}_{1-\lambda}(\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)) + \right. \\ \left. \begin{array}{l} (1 - \lambda) \overset{\leftrightarrow}{H} \left(\begin{array}{l} \tilde{N}_1(\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)); \\ p^I, 0 \end{array} \right) + \\ + \lambda \overset{\leftrightarrow}{H} \left(\begin{array}{l} \tilde{N}_0(\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)); \\ p^I, 1 \end{array} \right) \end{array} \right) \end{array} \right) \quad (70)$$

$$\begin{aligned}
 agr_r(\bar{\mathbf{x}}, \bar{y}^r) &= \\
 &= \left(\begin{array}{l} (1 - \alpha^{\text{agr}}) \text{avg} (I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \dots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r)) + \\ + \alpha^{\text{agr}} \left(\begin{array}{l} (1 - \lambda) \overset{\leftrightarrow}{H} \left(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \dots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); \right) + \\ \lambda \overset{\leftrightarrow}{H} \left(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \dots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); \right) \\ w_1^{\text{agr}}, \dots, w_N^{\text{agr}}, p^{\text{agr}}, 0 \end{array} \right) \end{array} \right) \quad (71)
 \end{aligned}$$

Learning procedures

We will explain how to learn parameters of membership functions and flexibility parameters in systems presented in sections “Flexible neuro-fuzzy systems”

and “Compromise neuro-fuzzy systems”. Let $\bar{x}(t) \in R^n$ and $d(t) \in R$ be a sequence of inputs and desirable output signals, respectively.

Based on the learning sequence $(\bar{x}(1), d(1)), (\bar{x}(2), d(2)), \dots$ we wish to determine all parameters (including the compromise parameter λ and system’s type ν) and weights of neuro-fuzzy systems such that

$$e(t) = \frac{1}{2} [f(\bar{x}(t)) - d(t)]^2 \quad (72)$$

is minimized, where $f(\cdot)$ is given by (59) or (65). The steepest descent optimization algorithm can be applied to solve this problem. For instance, parameters $\bar{y}^r, r = 1, \dots, N$, are trained by the iterative procedure

$$\bar{y}^r(t+1) = \bar{y}^r(t) - \eta \frac{\partial e(t)}{\partial \bar{y}^r(t)}. \quad (73)$$

Directly calculating partial derivatives in recursion (73) is rather complicated. Therefore, we recall that our system has a layered architecture (see Fig. 2) and apply the idea of the backpropagation method to train the system. The exact recursions are not shown here, however they can be derived analogously to the method given in [16]. We can apply the gradient optimization with constraints in order to optimize:

$$\begin{aligned} \nu &\in [0, 1], \quad \lambda \in [0, 1] \\ \alpha^T &\in [0, 1], \quad \alpha^I \in [0, 1], \quad \alpha^{\text{agr}} \in [0, 1], \\ p^T &\in [0, \infty), \quad p^I \in [0, \infty), \quad p^{\text{agr}} \in [0, \infty), \\ w_{i,k}^T &\in [0, 1], \quad i = 1, \dots, n, \quad k = 1, \dots, N, \\ w_{i,k}^{\text{agr}} &\in [0, 1], \quad k = 1, \dots, N. \end{aligned}$$

The same technique can be used in order to find in the process of learning parameters of the membership functions $\mu_{A_i^k}(x_i)$ and $\mu_{B^k}(y), i = 1, \dots, n, k = 1, \dots, N$.

Simulation results

In this section we apply neuro-fuzzy systems presented in sections “Flexible neuro-fuzzy systems” and “Compromise neuro-fuzzy systems” to identify a plant described by the difference equation (see [19])

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g(u(k)) \quad (74)$$

where the unknown function has the form

$$g(u) = 0.6 \sin(\pi u) + 0.3 \sin(3\pi u) + 0.1 \sin(5\pi u) \quad (75)$$

and the input is described by

$$u(k) = \sin\left(\frac{2\pi k}{250}\right) \quad (76)$$

In order to identify the plant, we use a model governed by the difference equation

$$\hat{y}(k+1) = 0.3y(k) + 0.6y(k-1) + \hat{f}(u(k)) \quad (77)$$

where $\hat{f}(\cdot)$ is a flexible neuro-fuzzy system presented in section 4 (called OR-type system) or a compromise neuro-fuzzy system presented in section 5 (called AND-type system).

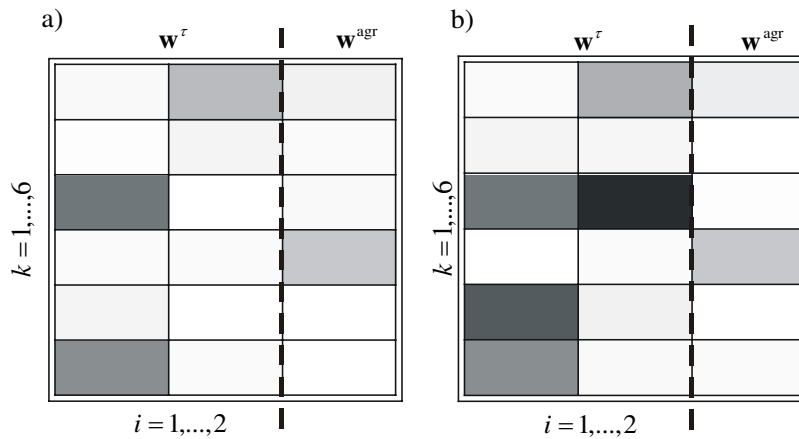


Figure 6. Weights representation in the Nonlinear Dynamic Plant problem for OR-type NFIS and a) Zadeh H -function, b) algebraic H -function

All the simulations are designed in the same fashion. We will gradually incorporate flexibility parameters in experiments (i)-(iv):

- In the first experiment (i), based on the input-output data, we learn the parameters of the membership functions and a system type $\nu \in [0, 1]$ or

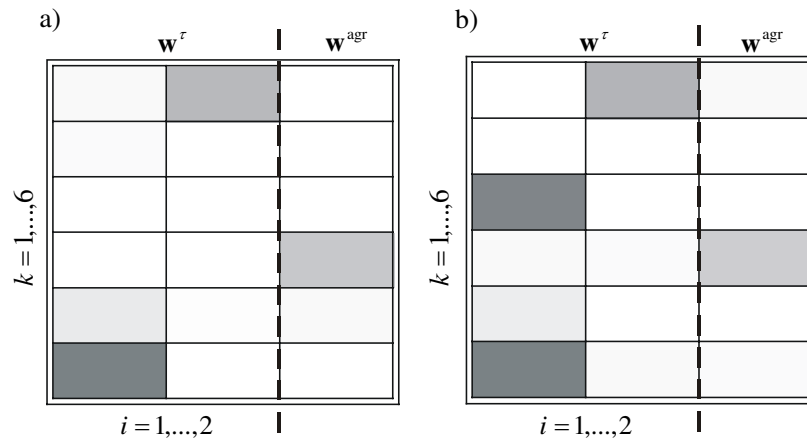


Figure 7. Weights representation in the Nonlinear Dynamic Plant problem for OR-type NFIS and a) Dombi H -function, b) Yager H -function

$\lambda \in [0, 1]$ assuming that there are no other flexibility parameters in the system description. It will be seen that the optimal values of ν or λ , determined by a gradient procedure, are either zero or one.

- In the second experiment (ii), we learn the parameters of the membership functions choosing value ν or λ as opposite to that obtained in experiment (i). Obviously, we expect a worse performance of the neuro-fuzzy system comparing with experiment (i).
- In the third experiment (iii), we learn the parameters of the membership functions, system type $\nu \in [0, 1]$ or $\lambda \in [0, 1]$ and soft parameters $\alpha^T \in [0, 1]$, $\alpha^I \in [0, 1]$, $\alpha^{\text{agr}} \in [0, 1]$ of the flexible system assuming that classical (not-parameterized) triangular norms are applied.
- In the fourth experiment (iv), we learn the same parameters as in the third experiment and, moreover, the weights $w_{i,k}^\tau \in [0, 1]$, $i = 1, \dots, n$, $k = 1, \dots, N$, in the antecedents of rules and weights $w_{i,k}^{\text{agr}} \in [0, 1]$, $k = 1, \dots, N$, of the aggregation operator of rules. In all diagrams (weights representation) we separate $w_{i,k}^\tau \in [0, 1]$, $i = 1, \dots, n$, $k = 1, \dots, N$, from $w_{i,k}^{\text{agr}} \in [0, 1]$, $k = 1, \dots, N$, by a vertical dashed line. In the

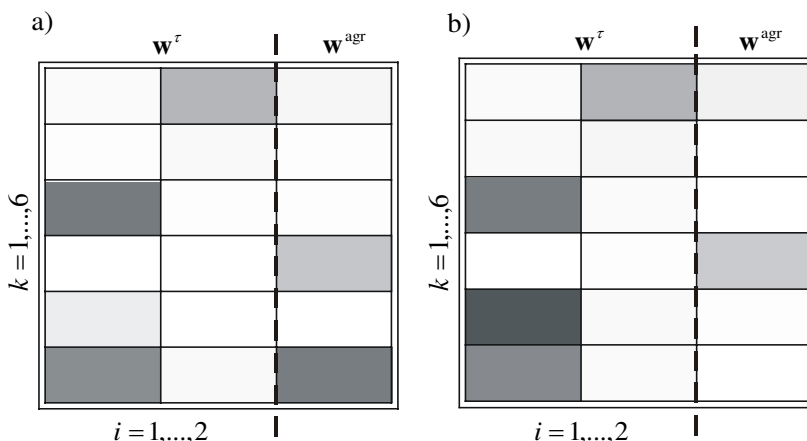


Figure 8. Weights representation in the Nonlinear Dynamic Plant problem for AND-type NFIS and a) Zadeh H -function, b) algebraic H -function

diagrams dark areas correspond to low values of weights and vice versa.

In each of the above simulations we apply the Zadeh H -implication (generated by the min/max triangular norms) and the algebraic H -implication (generated by the algebraic triangular norms). In separate experiments we repeat simulations (i)–(iv) replacing the Zadeh H -implication and the algebraic H -implication by quasi-implications generated by parameterized triangular norms: the Dombi H -implication and the Yager H -implication. In these simulations we additionally incorporate parameters $p^\tau \in [0, \infty)$, $p^I \in [0, \infty)$, $p^{agr} \in [0, \infty)$.

a) Flexible neuro-fuzzy systems

We apply neuro-fuzzy systems described in section 4 and given by formulas (59)–(62). The experimental results for the Nonlinear Dynamic Plant problem (72) are depicted in Tables 1 and 2 for the not-parameterized (Zadeh and algebraic) and parameterized (Dombi and Yager) H -functions, respectively. For experiment (iv) the final values (after learning) of weights $w_{i,k}^\tau \in [0, 1]$ and $w_k^{agr} \in [0, 1]$, $i = 1, \dots, 2$, $k = 1, \dots, 6$, are shown in Fig. 6 (Zadeh and algebraic H -functions) and Fig. 7 (Dombi and Yager H -functions).

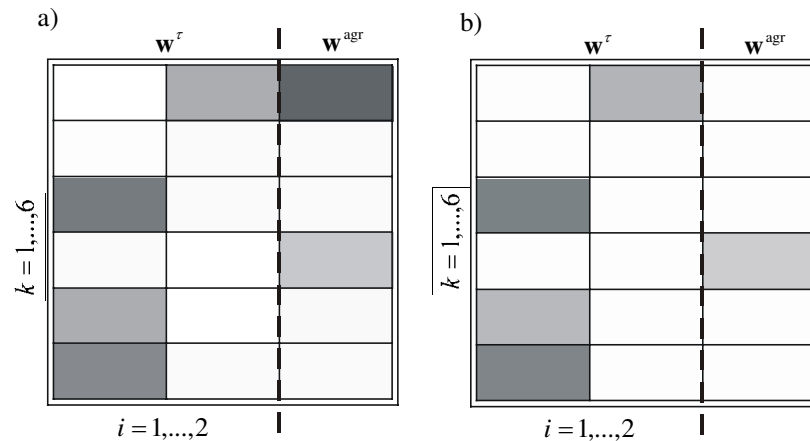


Figure 9. Weights representation in the Nonlinear Dynamic Plant problem for AND-type NFIS and a) Dombi H -function, b) Yager H -function

b) Compromise neuro-fuzzy systems

We apply neuro-fuzzy systems described in section 5 and given by formulas (65)–(68). The experimental results for the Nonlinear Dynamic Plant problem (72) are depicted in Tables 3 and 4 for the not-parameterized (Zadeh and algebraic) and parameterized (Dombi and Yager) H -functions, respectively. For experiment (iv) the final values (after learning) of weights $w_{i,k}^{\tau} \in [0, 1]$ and $w_k^{\text{agr}} \in [0, 1]$, $i = 1, \dots, 2$, $k = 1, \dots, 6$, are shown in Fig. 7 (Zadeh and algebraic H -functions) and Fig. 8 (Dombi and Yager H -functions).

It should be noted that flexible and compromise neuro-fuzzy systems perfectly approximate the plant. For example in Fig. 10 we show signals $y(k)$ and $\hat{y}(k)$, applying system given by (59)–(62), after 4000 epochs. From tables 1 and 2 it follows that the flexible neuro-fuzzy system (59)–(62) becomes of a Mamdani-type ($\nu = 0$) when the learning process is completed. The same conclusion ($\lambda = 0$) is valid for compromise neuro-fuzzy system (65)–(68). In other investigations (see [12,16,17]) it was found out that neuro-fuzzy systems become of a logical-type for classification problems.

Table 1. Experimental results — system given by (59)–(62) and non-parametrized H -functions

OR-TYPE NFIS WITH NON-PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)	
			Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	ν	0.5	0.0000	0.0000	0.0468	0.0311
ii	ν	1	-	-	0.0562	0.0370
iii	ν	0.5	0.0000	0.0000	0.0384	0.0291
	α^r	1	0.8012	0.9896		
	α^l	1	0.7313	0.9964		
	α^{agr}	1	0.9936	0.9812		
iv	ν	0.5	0.0000	0.0000	0.0271	0.0168
	α^r	1	0.8331	0.9925		
	α^l	1	0.6991	0.9968		
	α^{agr}	1	0.9959	0.9847		
	w^r	1	Fig. 1-a	Fig. 1-b		
	w^{agr}	1	Fig. 1-a	Fig. 1-b		

Table 2. Experimental results — system given by (59)–(62) and parametrized H -functions

OR-TYPE NFIS WITH PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)	
			Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	ν	0.5	0.0000	0.0000	0.0392	0.0343
ii	ν	1	-	-	0.0437	0.0312
iii	ν	0.5	0.0000	0.0000	0.0304	0.0248
	p^r	10	9.9986	10.9834		
	p^l	10	8.6509	11.9798		
	p^{agr}	10	12.8795	7.6195		
	α^r	1	0.8218	0.9954		
	α^l	1	0.8913	0.9514		
iv	ν	0.5	0.0000	0.0000	0.0202	0.0176
	p^r	10	9.1298	9.9465		
	p^l	10	7.0674	15.2263		
	p^{agr}	10	11.0981	4.1546		
	α^r	1	0.8694	0.9947		
	α^l	1	0.7943	0.9273		
	α^{agr}	1	0.9896	0.9187		
	\mathbf{w}^r	$\mathbf{1}$		Fig. 2-a		
\mathbf{w}^{agr}	$\mathbf{1}$		Fig. 2-a	Fig. 2-b		

Table 3. Experimental results — system given by (65)–(68) and non-parametrized H -functions

AND-TYPE NFIS WITH NON-PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)	
			Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	λ	0.5	0.0000	0.0000	0.0361	0.0342
ii	λ	1	-	-	0.0616	0.0406
iii	λ	0.5	0.0000	0.0000	0.0279	0.0216
	α^r	1	0.8570	0.9891		
	α^l	1	0.6453	0.9839		
	α^{agr}	1	0.9724	0.9952		
iv	λ	0.5	0.0000	0.0000	0.0213	0.0137
	α^r	1	0.8645	0.9847		
	α^l	1	0.7957	0.9587		
	α^{agr}	1	0.9834	0.9893		
	\mathbf{w}^r	1	Fig. 3-a	Fig. 3-b		
\mathbf{w}^{agr}	1	Fig. 3-a	Fig. 3-b			

Table 4. Experimental results – system given by (65)–(68) and parametrized H -functions

AND-TYPE NFIS WITH PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)							
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)		
			Dombi H-function	Yager H-function	Dombi H-function	Yager H-function	
i	λ	0.5	0.0000	0.0000	0.0391	0.0365	
ii	λ	1	-	-	0.0645	0.0578	
iii	λ	0.5	0.0000	0.0000	0.0266	0.0274	
	p^r	10	9.6428	10.9448			
	p^l	10	7.9106	13.8753			
	p^{agr}	10	10.2587	3.1987			
	α^r	1	0.8927	0.9817			
	α^l	1	0.7314	0.9678			
	α^{agr}	1	0.9927	0.9824			
iv	λ	0.5	0.0000	0.0000	0.0207	0.0186	
	p^r	10	10.0024	9.5846			
	p^l	10	7.0568	10.8733			
	p^{agr}	10	11.5201	6.5864			
	α^r	1	0.8942	0.9817			
	α^l	1	0.8917	0.9439			
		α^{agr}	1	0.9949			0.9486
		w^r	1	Fig. 4-a			Fig. 4-b
		w^{agr}	1	Fig. 4-a			Fig. 4-b

Final remarks: Design of flexible neuro-fuzzy systems

In the paper we introduced several flexibility concepts in the design of neuro-fuzzy systems:

- softness to fuzzy implication operators, to the aggregation of rules and to the connectives of antecedents,
- certainty weights to the aggregation of rules and to the connectives of antecedents,
- parameterized families of t -norms and t -conorms to fuzzy implication operators, to the aggregation of rules and to the connectives of antecedents.

Moreover, our design process was characterized by the automatic determination of fuzzy inference described by parameter ν or λ . The main advantage of our approach is a high accuracy of neuro-fuzzy systems. In the future research we plan to apply neuro-fuzzy systems in a time-varying environment [13]–[15].

References

1. *Chen M.Y., Linkens D.A.* A systematic neuro-fuzzy modeling framework with application to material property prediction // *IEEE Trans. on Fuzzy Systems*, **31** (2001) 781–790.
2. *González, Pérez R.* SLAVE: A genetic learning system based on an iterative approach // *IEEE Trans. on Fuzzy Systems*, **7** (1999) 176–191.
3. *Jang J.S., Sun C.T., Mizutani E.* Neuro-Fuzzy and Soft Computing, Prentice Hall, Englewood Cliffs 1997.
4. *Kasabov N.* DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction // *IEEE Trans. on Fuzzy Systems*, **10** (2002) 144–154.
5. *Kim E., Park M., Ji S., Park M.* A new approach to fuzzy modeling // *IEEE Trans. on Fuzzy Systems*, **5** (1997) 328–337.
6. *Klement E.P., Mesiar R., Pap E.* Triangular Norms, Kluwer Academic Publishers, Netherlands 2000.
7. *Roubos H., Setnes M.* Compact and transparent fuzzy models and classifiers through iterative complexity reduction // *IEEE Trans. on Fuzzy Systems*, **9** (2001) 516–524.
8. *Rutkowski L., Cpalka K.* Flexible Structures of Neuro-Fuzzy Systems, Quo Vadis Computational Intelligence // *Studies in Fuzziness and Soft Computing*, Springer-Verlag, **54** (2000) 479–484.

9. Rutkowski L., Cpalka K. A general approach to neuro-fuzzy systems // *Proc. of the 10th IEEE Intern. Conference on Fuzzy Systems*, Melbourne 2001.
10. Rutkowski L., Cpalka K. A neuro-fuzzy controller with a compromise fuzzy reasoning // *Control and Cybernetics*, **31**, 2 (2002) 297–308.
11. Rutkowski L., Cpalka K. Flexible weighted neuro-fuzzy systems // *Proc. 9th Intern. Conference on Neural Information Processing (ICONIP'02)*, Orchid Country Club, Singapore 2002.
12. Rutkowski L., Cpalka K. Flexible neuro-fuzzy systems // *IEEE Trans. Neural Networks*, **14** (May 2003) 554–574.
13. Rutkowski L. Adaptive probabilistic neural networks for pattern classification in time-varying environment // *IEEE Trans. on Neural Networks*, **15** (2004) 811–827.
14. Rutkowski L. Generalized regression neural networks in time-varying environment // *IEEE Trans. on Neural Networks*, **15** (2004) 576–596.
15. Rutkowski L. *New Soft Computing Techniques For System Modeling, Pattern Classification and Image Processing*, Springer-Verlag, 2004.
16. Rutkowski L. *Flexible Neuro-Fuzzy Systems*, Kluwer Academic Publishers, 2004.
17. Rutkowski L. A new method for system modelling and pattern classification // *Bulletin of the Polish Academy of Sciences*, **52** (2004) 11–24.
18. Setnes M., Roubos H. GA-fuzzy modeling and classification complexity and performance // *IEEE Trans. on Fuzzy Systems*, **8** (2000) 509–521.
19. Wang L. *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*, Prentice Hall, 1994.
20. Wang L., Yen J. Application of statistical information criteria for optimal fuzzy model construction // *IEEE Trans. on Fuzzy Systems*, **6** (1998) 362–371.
21. Yager R.R., Filev D.P. *Essentials of Fuzzy Modeling and Control*, New York, John Wiley & Sons, 1994.

Leszek RUTKOWSKI was born in Wroclaw, Poland, in 1952. He received M. Sc., Ph. D. and D. Sc. degrees in 1977, 1980, 1986, respectively, all from the Technical University of Wroclaw, Poland. Since 1980, he has been with the Technical University of Czestochowa where he is currently a Professor and Chairman of the Computer Engineering Department. From 1987 to 1990 he held a visiting position in the School of Electrical and Computer Engineering at Oklahoma State University. His research interests include neural networks, fuzzy systems, computational intelligence, pattern recognition and systems identification. He published over 100 technical papers including 17 in various series of

IEEE Transactions. He is the author of the books “New Soft Computing Techniques For System Modelling, Pattern Classification and Image Processing” published by Springer, “Flexible Neuro-Fuzzy Systems” published by Kluwer Academic Publishers, “Adaptive Filters and Adaptive Signal Processing” (in Polish), and co-author of two others (in Polish and Russian) “Neural Networks, Genetic Algorithms and Fuzzy Systems” and “Neural Networks for Image Compression”. He is also President and Founder of the Polish Neural Networks Society. He organized and served as General Chair of the Polish Neural Networks Conferences held in: 1996, 1997, 1999, 2000, 2002. Prof. Leszek Rutkowski is an Associate Editor of the IEEE Transactions on Neural Networks and recently elected member of the Polish Academy of Sciences.

Лешек РУТКОВСКИЙ родился во Вроцлаве, Польша, в 1952 г. Степени M. Sc., Ph. D. и D. Sc. получил 1977, 1980, 1986, соответственно, все во Вроцлавском техническом университете. Начиная с 1980 года он работает в Ченстоховском техническом университете, Ченстохова, Польша, где является в настоящее время профессором и заведующим кафедрой вычислительной техники. С 1987 по 1990 год он работал в качестве приглашенного профессора на факультете электротехники и вычислительной техники Оклахомского государственного университета, Оклахома, США. Исследовательские интересы проф. Рутковского включают нейронные сети, нечеткие системы, вычислительный интеллект, распознавание образов и идентификацию систем. Им опубликовано свыше 100 научных работ, включая 17 статей в различных сериях журнала IEEE Transactions. Проф. Рутковский является автором нескольких книг, в том числе: «Новые методы мягких вычислений для моделирования систем, классификации образов и обработки изображений» (издательство Springer, на английском языке), «Гибкие нейро-нечеткие системы» (издательство Kluwer Academic Publishers, на английском языке), «Адаптивные фильтры и адаптивная обработка сигналов» (издано на польском языке), а также соавтором книг «Нейронные сети, генетические алгоритмы и нечеткие системы» (издано на польском и русском языках) и «Нейронные сети для сжатия изображений». Проф. Рутковский является членом Польской Академии наук, основателем и президентом Польского общества нейронных сетей. Он был председателем оргкомитетов Польских конференций по нейронным сетям, проводившихся в 1996, 1997, 1999, 2000 и 2002 годах. Проф. Рутковский входит в состав редакционной коллегии журнала IEEE Transactions on Neural Networks.

Н. Г. ЯРУШКИНА

Ульяновский государственный технический университет

УлГТУ, г. Ульяновск,

E-mail: jng@ulstu.ru

**НЕЧЕТКИЕ НЕЙРОННЫЕ СЕТИ В КОГНИТИВНОМ
МОДЕЛИРОВАНИИ И ТРАДИЦИОННЫХ ЗАДАЧАХ
ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

Аннотация

Нечеткие нейронные сети (ННС) представляют собой реализацию систем нечеткого логического вывода методами нейронных сетей. Рассмотрен когнитивный потенциал ННС для широкого спектра задач: соотношения логического и образного в мышлении, осознанного и неосознанного, обобщенного и конкретного. Задачи использования когнитивных моделей в искусственных системах решает традиционный искусственный интеллект. В лекции сопоставлены задачи инженерии знаний и возможности нейросетевых технологий. Приведены примеры использования ННС для создания прикладных интеллектуальных систем.

N. G. YARUSHKINA

Ulyanovsk State Technical University,

UISTU, Ulyanovsk,

E-mail: jng@ulstu.ru

**FUZZY NEURON NETWORKS IN COGNITIVE MODELLING AND
ARTIFICIAL INTELLIGENCE**

Abstract

Fuzzy neuron networks (FNN) with genetic optimization are fuzzy logic inference systems. The FNNs have cognitive possibilities for development of different human brain functions models. A basic result of Artificial Intelligence is Knowledge Engineering. In this Lecture knowledge engineering and possibilities of the FNN are compared. An example is considered for some kind of applied intelligence system.

Введение

В современной нейроинформатике соотношение фундаментальных и технологических задач складывается в пользу технологических. Отметим, что фундаментальные задачи — моделирование ощущений, восприятий, распознавание образов, обучение и запоминание паттернов являются основными задачами когнитивной науки. Технологические задачи связаны с повышением эффективности использования нейронных сетей в решении задач кластеризации, классификации образов и сцен, аппроксимации функций для процессов технической, экономической, биологической природы. Достижения нейроинформатики в решении технологических задач привели к созданию целого спектра разнообразных нейронных сетей: сетей с прямым распространением сигнала, рекуррентных сетей, радиально базисных сетей. Созданы развитые архитектуры гибридных систем: нечетких нейронных сетей, нечетких нейронных с генетической настройкой параметров.

Ситуация в нейроинформатике требует вернуться к фундаментальным когнитивным задачам и выяснить, позволяют ли новые классы нейронных и гибридных систем решать новые когнитивные задачи. Далее рассматриваются в основном нечеткие нейронные сети с генетической оптимизацией параметров, так как они обладают комплексной структурой и сложными алгоритмами обучения, интегрирующими структурные части различных нейронных сетей. Следовательно, именно гибридные системы, как можно предположить, должны обладать сильным когнитивным потенциалом. Далее будем называть именно их гибридными системами. Гибридные системы включают в себя слои радиально базисных нейронов, логических нейронов, традиционных пороговых суммирующих нейронов. Алгоритм их обучения обычно комбинирует соревновательное обучение (по алгоритму победителя), генетическую оптимизацию параметров и классический метод обратного распространения ошибки.

Рассмотрим также некоторые новые применения гибридных интеллектуальных систем, характеризующие их прикладной потенциал, в том числе для традиционных задач искусственного интеллекта. Выберем направления, сочетающие распознавание и логическую обработку. На наш взгляд место для успешных применений гибридных технологий может быть найдено в направлении интеллектуализации информационных систем: баз данных и корпоративных хранилищ ресурсов.

Интеллектуализация баз данных

Технология баз данных является важнейшей из прикладных технологий информационных систем, так как на ней строятся экономические информационные системы любого класса. Потребность в интеллектуализации баз данных связана с несколькими причинами.

Базы данных состоят из десятков тысяч рабочих таблиц, индексов, ограничений, доменов, представлений. Простого словаря, регистрирующего объекты, недостаточно для управления данными. Обработка транзакций в корпоративных системах (более 1000 автоматизированных рабочих мест) требует встраивания в сервер данных логики событий. Состоявшаяся в свое время в искусственном интеллекте широкая дискуссия о том, чем данные отличаются от знаний, остановилась на мнении, что «знания, это данные и их интерпретация», то есть знания — это еще и метазнания. Ограничения целостности, характерные для серверов данных, это всегда знания о законах проблемной области. Показательно, что крупные производители систем баз данных (Oracle Inc.) включают в состав программного продукта систему описания и интерпретации бизнес-правил в форме продукций ЕСЛИ–ТО и систему логического вывода действий. Любое действие, которое выводится из бизнес-правил, реализуется в системе как транзакция обработки данных.

Новые возможностные типы данных

Взаимодействие баз данных с системами обработки знаний расширяет базовые типы данных реляционной модели данных. Если система знаний представлена системой нечетких продукций, то такой системе становится необходимой среда хранения и обработки нечетких атрибутов. В настоящее время построена нечеткая реляционная алгебра, выполняющая операции соединения, селекции, проекции, объединения и пересечения. Нечеткий реляционный сервер данных — это репозиторий, который предназначен для хранения функций принадлежности, лингвистических меток.

Интеллектуальный анализ данных Data Mining

Анализ данных (Data Mining) — это научное направление, изучающее методы извлечения законов проблемной области из баз данных. Такие «законы»,

разумеется, не имеют строгой аналитической формы, в лучшем случае аппроксимируются некоторым набором правил ЕСЛИ–ТО. Следовательно, Data Miner имеет на входе базу данных (обучающую выборку), а на выходе базу правил (базу знаний). Но именно такие входы и выходы имеет нечеткая нейронная сеть, которая в таком контексте становится средством класса Data Miner.

Интеллектуализация Intranet и Internet-технологий

Корпоративные хранилища информационных ресурсов строятся на основе технологии менеджеров файлов и внутреннего WWW-сервиса. Под ресурсом понимают совокупность файлов, объединенных общей семантикой. Хранение и использование таких ресурсов требует построения индексов хранения, которые не просто обеспечивают быстрый поиск ресурса, а становятся метаописанием ресурсов, т. е. знаниями.

Интеллектуальная поисковая машина ресурсов

Чтобы обеспечить хранение и поиск ресурсов, необходимо формировать индексы (каталоги) в соответствии с систематическими рубриками. Рубрики представляют собой классы объектов проблемной области, причем трудно вручную составить исходную достаточно полную классификацию ресурсов. Следовательно, необходимо решить задачу кластеризации — порождение системы нечетких классов для ресурсов. Затем, каждый новый ресурс необходимо относить к нечетким классам. Очевидно, что такую задачу можно решить только на основе аннотаций текстов, которые должны присутствовать в каждом ресурсе. Текст можно отнести к нечеткому классу на основе встречаемости ключевых слов. Для решения задачи кластеризации можно применить нейронную сеть Кохонена, обучаемую по алгоритму победителя, задача классификации ресурсов решается с помощью нечеткой нейронной сети, например, класса ANFIS.

Нечеткие нейронные сети

Определение нечеткого множества

Нечетким множеством f называют множество базовых элементов b , степень принадлежности которых выражается с помощью функции принадлежности, обозначаемой μ . Функция $\mu(b, f)$ отображает базовое значение b и нечеткое значение f в интервал $[0, 1]$. По определению, $0 < \mu(b, f) < 1$ для b и f . На рис. 1 приведены нечеткие значения, характерные для переменной *скорость автомобиля*.

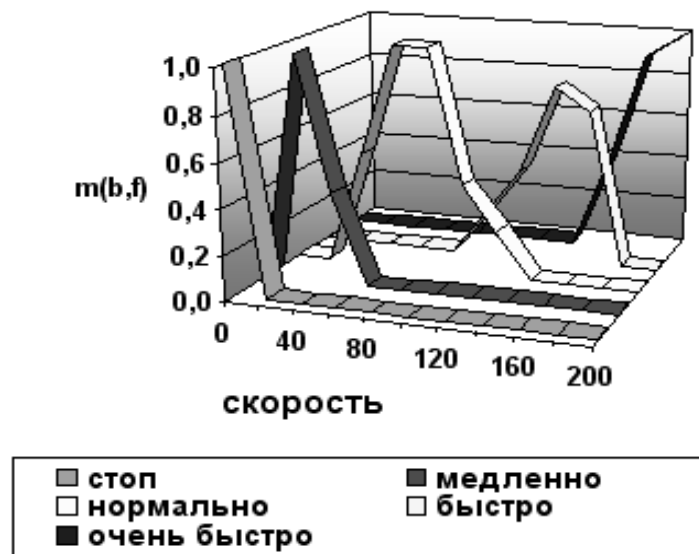


Рис. 1. Пример определения нечетких значений скорости автомобиля

Определение нечеткой системы

Системы нечетких продукций строятся на основе понятия лингвистической переменной, которой называют пятерку объектов:

$$\langle x, T(x), U, G, M \rangle,$$

где x — собственное имя переменной; $T(x)$ — терминальное множество, то есть набор значений переменной (нечетких меток); U — множество объектов (или универсум); G — синтаксические правила употребления; M — семантические правила употребления.

Обобщив нечеткую зависимость переменных x и y , получим продукцию: **если** $A(x)$, **то** $B(y)$. Цель нечеткого логического вывода — определить значения выходной переменной $B'(y)$ в зависимости от значений входной переменной $C(x')$.

Основными правилами умозаключений являются два силлогизма: *modus ponens* (MP) и *modus tollens* (MT). Рассмотрим изменение основных правил вывода в случае, если предпосылка и заключение выражаются в терминах нечетких множеств:

Нечеткий *modus ponens* (*Fuzzy MP*):

если $A(x)$ **то** $B(x)$; $A'(x)$, **то** $B'(x)$.

Нечеткий *Modus tollens* (*FMT*):

если $A(x)$ **то** $B(x)$, $\neg B'(x)$, **то** $\neg A'(x)$.

Системы нечеткого вывода

Схема вывода на базе нечетких правил сводится к решению следующей задачи:

База правил	R_1 : если $(X - A_1)$ И $(Y - B_1)$ тогда $(Z - C_1)$ в противном случае
	R_2 : если $(X - A_2)$ И $(Y - B_2)$ тогда $(Z - C_2)$ в противном случае
	...
	R_n : если $(X - A_n)$ И $(Y - B_n)$ тогда $(Z - C_n)$
Факт:	$x = x_0, y = y_0$.
Следствие:	$z = ?$

Интеллектуальная система осуществляет логический вывод по базе нечетких правил по этапам.

- Фаззификация фактических данных, т. е. точное значение x_0 интерпретируется как нечеткая точка.
- Композиция входной переменной и условной части правила: $x_0 \circ A_i, y_0 \circ B_i$, т. е. вычисляется уровень пригодности правила к

ситуации. Если факт задан нечеткой точкой, то композиция сводится к выявлению соответствующей степени принадлежности.

- Вычисление нечеткой импликации

$$((x_0 \circ A_i) \cap (y_0 \circ B_i)) \rightarrow C_i$$

для $\forall R$. Результатом выполнения для всех правил являются N нечетких значений для выхода Z .

- Агрегация среднего значения, т. е. построение нечеткого значения выхода по результатам предыдущих этапов:

$$C = \bigcup_{i=1}^n C_i.$$

- Дефаззификация, т. е. выбор представляющего элемента по агрегированному нечеткому понятию.

Определения И- и ИЛИ-нейронов

Гибридные системы, которые в качестве базовой используют нейронные сети, интерпретируемые как системы нечеткого вывода, называются нечеткими нейронными сетями. Рассмотрим далее нечеткие нейронные сети, как пример успешной технологии вычислительного интеллекта. Глубинная интеграция нечетких систем и нейросетей, связана с разработкой новой архитектуры элементов нейросети. Для интеграции двух технологий — нечетких систем и нейрокомпьютинга, необходимо предложить способ четкого дискретного представления непрерывных функций принадлежности, для чего выберем максимально большой интервал $[x_1, x_2]$, в котором представлены все нечеткие множества условных частей правил. Если разбить интервал с равным шагом, то любое нечеткое значение представляется четким вектором. Другой способ представления нечеткого понятия в виде четких данных состоит в представлении нечеткого множества в виде совокупности α -срезов (рис. 2).

При использовании α -срезов каждое α_j -подмножество представляется двумя числами — левой и правой границами: $\alpha - L_{ij}$, $\alpha - R_{ij}$, где j — номер α -среза, а i — номера точек на его левой и правой границах, т. е. α -срезы четко представляют непрерывную функцию принадлежности. Модификация модели нейрона для адаптации к нечетким системам касается выбора

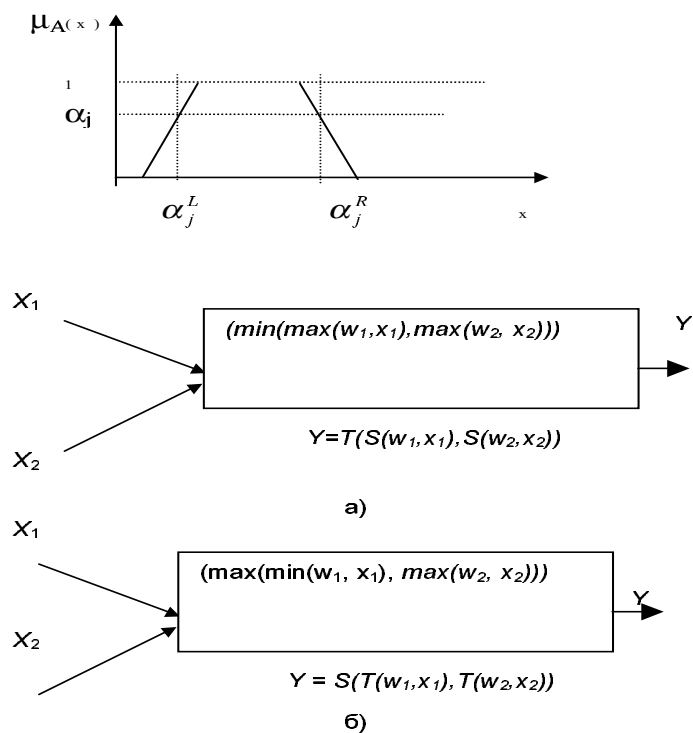


Рис. 2. α -срез, И-ИЛИ нейроны

функции активации, реализации операций сложения и умножения, так как в нечеткой логике сложение моделируется любой треугольной конормой (например, $\max, a + b - a * b, \dots$), а операция умножения — треугольной нормой ($\min, a * b, \dots$).

И-нейроном называется нейрон, в котором умножение веса w на вход x моделируется конормой $S(w, x)$, а сложение — нормой $T(w, x)$.

Для двухвходового И-нейрона справедлива формула:

$$Y = T[S(w_1, x_1), S(w_2, x_2)].$$

ИЛИ-нейроном называется нейрон, в котором умножение веса w и

входа x моделируется нормой $T(w, x)$, а сложение взвешенных весов — конормой $S(w, y)$. Для двухвходового *ИЛИ*-нейрона справедлива формула:

$$Y = S[T(w_1, x_1), T(w_2, x_2)].$$

Если выбрать в качестве T -нормы \min , а \max — в качестве S -нормы, то формула преобразования *ИЛИ*-нейрона уточняется следующим образом:

$$\max[\min(w_1, x_1), \min(w_2, x_2)].$$

В качестве функции активации обычно используют радиальную базисную функцию $F(x) = \exp[-b * (x^2 - a)]$.

Определение нечеткой нейронной сети

Нечеткой нейронной сетью (НС) [5] обычно называют четкую нейросеть, которая построена на основе многослойной архитектуры с использованием *И*-, *ИЛИ*-нейронов. Нечеткая нейросеть функционирует стандартным образом на основе четких действительных чисел. Нечеткой является только интерпретация результатов. При создании гибридной технологии кроме объединения систем по данным можно использовать нейрокомпьютинг для решения частной подзадачи нечетких экспертных систем, а именно настройки параметров функции принадлежности. Функции принадлежности можно сформировать двумя способами: методом экспертной оценки; на основе статистики. Гибридные технологии предлагают третий способ: в качестве функции принадлежности выбирается параметризованная функция формы (например, гауссова кривая с параметрами b, a), параметры которой настраиваются с помощью нейросетей. Настройка параметров может быть получена в рамках алгоритма обратного распространения ошибки. Таким образом, нечеткая нейронная сеть наряду с классическими нейронами, являющимися пороговыми суммирующими элементами, должна включать в себя *И*-, *ИЛИ*-нейроны. В качестве примера рассмотрим архитектуру нечеткой нейронной сети класса ANFIS, включающую адаптивный слой функций принадлежности и логические *И*-, *ИЛИ*-нейроны, «логические» нейроны, моделирующие логические связи.

ANFIS — адаптивная НС, основанная на системе нечеткого вывода

Приведем на рис. 3 структуру ANFIS (Adaptive Network based Fuzzy Inference System) — адаптивной НС для двух правил:

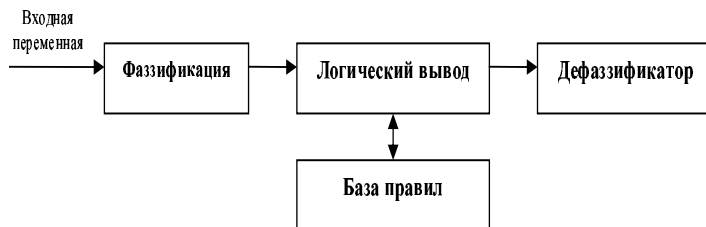
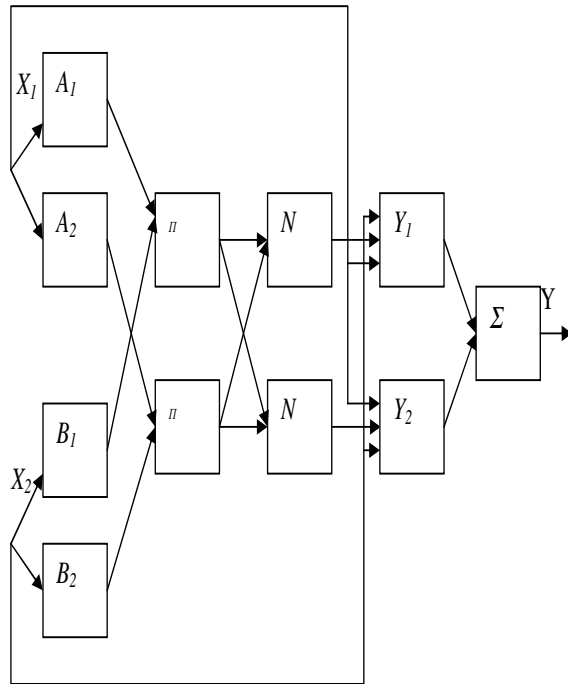


Рис. 3. Структуры ANFIS и нечеткой экспертной системы

если $x_1 = A_1$ **И** $x_2 = B_1$ **то** $y_1 = c_{11} * x_1 + c_{12} * x_2$

если $x_1 = A_2$ **И** $x_2 = B_2$ **то** $y_2 = c_{21} * x_1 + c_{22} * x_2$

Выход ННС формируется по формуле:

$$y = \frac{(w_1 * y_1 + w_2 * y_2)}{(w_1 + w_2)}.$$

Слой ННС ANFIS выполняют следующие функции.

Слой 1 представлен радиальными базисными нейронами и моделирует функции принадлежности.

Слой 2 — это слой *И*-нейронов, которые моделируют логическую связку *И* произведением $w_i = \mu_{A_i}(x_1) * \mu_{B_i}(x_2)$.

Слой 3 вычисляет нормированную силу правила:

$$w_i = \frac{w_i}{(w_1 + w_2)}.$$

Слой 4 формирует значение выходной переменной:

$$y(x_1, x_2) = w_i y_i = w_i (c_{i1} * x_1 + c_{i2} * x_2).$$

Слой 5 выполняет дефаззификацию:

$$y = w_1 * y_1 + w_2 * y_2.$$

Гибридная сеть архитектуры ANFIS обучается с помощью алгоритма обратного распространения ошибки.

Задачи когнитивного моделирования: Возможности ННС

Цели и задачи когнитивной науки

Исследование известных архитектур гибридных систем и алгоритмов их обучения позволяет, по нашему мнению, считать их моделями, адекватно отражающими соотношение восприятия и логического умозаключения при когнитивной деятельности человека. При реальном решении задачи человек комбинирует процессы мышления (компьютерный аналог — логический вывод), вспоминания (сознательное «внутреннее» возбуждение образа), восприятия окружающих предметов (распознавание) и, возможно,

движения (управление телом). Обученная нечеткая нейронная сеть хранит паттерны и выполняет логические операции, так как содержит *И-ИЛИ*-нейроны; она переключается с распознавания на логический вывод, так как обычно включает в себя несколько отдельно обученных сетей с разными функциями. Таким образом, несмотря на то, что нечеткие нейронные сети созданы как нейро-нечеткие контроллеры для управления процессами, они могут служить когнитивной моделью для изучения взаимодействия процессов восприятия и мышления при когнитивной деятельности.

Задача моделирования человеческого восприятия

Особенность человеческого восприятия — это способность понимать искаженную информацию (ассоциативная память). Глядя на образ, мы выделяем примитивы, которые порождают параметр порядка. В результате процесса самоорганизации и при подчинении параметру порядка человеческий мозг воссоздает целый образ и распознает объект. На основе теории синергетики (нелинейной динамики) создан синергетический компьютер в форме алгоритма [1]. Синергетический компьютер — это программа распознавания человеческих лиц, основанная на принципах самоорганизации. Такой компьютер проверен на искажениях, амбивалентных изображениях.

Синергетический компьютер

Детерминированный хаос — стереотип поведения многих синергетических систем, например, игровые автоматы часто используют запланированный хаос. Как хаос связан с синергетикой? Синергетическая система может управлять не одним а сразу несколькими параметрами порядка. Параметры сотрудничают, конкурируют, один из них может доминировать, но смена доминанты может быть хаотической. Следовательно, хаотическими являются те процессы, которые при малейшем изменении условий полностью изменяются. Хаос обладает двумя особенностями:

- чувствительность к исходным условиям;
- самоподобие.

Аттрактор — недостижимая точка притяжения состояния системы. Почему аттракторные НС имеют перспективы в решении интеллектуальных

задач? Из детерминированного и микроскопического хаоса возникает порядок. Предъявленный сенсорам образ, организует НС и позволяет извлечь нужный паттерн.

Например, рассмотрим применение аттракторной нейронной сети в анализе временных рядов (рис. 4). Необходима реконструкция аттракторов из временного ряда. В некоторый момент времени t_1 определено значение x , второе значение определяется в момент, смещенный относительно первого на T . Таким образом, обрабатываются все точки ряда, строится траектория на плоскости $X(t_1), X(t_1 + T)$. По траектории можно проследить расположение аттрактора, т. е. выполнить экстраполяцию. В случае хаотических аттракторов для его реконструкции требуется, по меньшей мере, трехмерная система координат (а то и большей размерности). Соответствующие новые координаты получают при этом смещением временной оси не только T , но и $2T$ и т. д. Проблема выбора смещения T не решена и определяется эмпирически.

Задача исследования баланса право- и левополушарных методов

В литературе встречаются схемы гибридизации нейроинформатики и ИИ, построенные по следующему принципу: правое полушарие — нейрокомпьютер; левое полушарие — основанная на знаниях система, а вопрос лишь в их взаимодействии или балансе право- и лево-полушарности. Причем гибридизация является именно глубинной, так как одна нейронная сеть содержит и логические (дискретные) *И-ИЛИ*-нейроны, и непрерывные традиционные пороговые суммирующие нейроны. Таким образом происходит создание структуры, объединяющей обработку сигнала и оценку такой обработки, в том числе логическую.

ННС — сочетание образного восприятия и логической обработки

В реальном поведении человека невозможно разделить восприятие и логическую обработку, поэтому более успешной представляется схема глубинной интеграции. Примерами таких глубинных успешных гибридов может служить создание нечетких нейронных сетей, объединяющих методы обучения нейронных сетей, управление неопределенностью нечеткой логики и логический вывод дедуктивных систем.

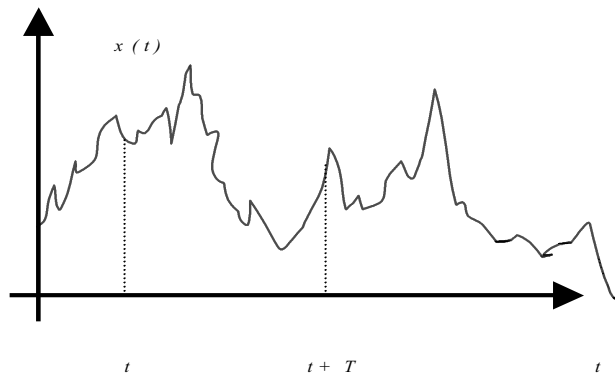
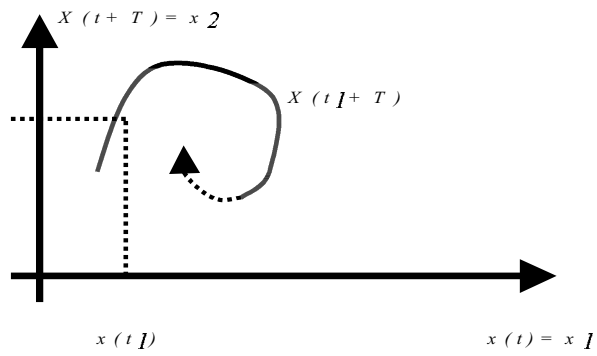


Рис. 4. Применение аттракторной НС для прогнозирования временно-го ряда

Задача самоорганизации знаний

Интеграция НС и вероятностных вычислений привела к созданию и широкому использованию радиальных базисных сетей, сетей регрессии, вероятностных сетей, байесовских сетей. Первой особенностью таких сетей является параллельное формирование не только решения, но и вероятностной оценки такого решения. Например, сети регрессии, которые используются в основном для классификации, не только соотносят с классом входной вектор, но и формируют вероятность такого классифицирования. Таким образом, вероятностные нейронные сети сочетают собственно распознавание образа с формированием его оценки и могут служить для моделирования восприятия. Второй особенностью вероятностных НС служит сложная послойная структура, так как разные слои содержат разные типы нейронов. Такая структура приводит к сложным многошаговым алгоритмам обучения, которые часто включают в себя соревновательное обучение по Хохену, обучение на основе штрафов, по алгоритму обратного распространения ошибки. Такие алгоритмы обучения далеки от минимизации функции ошибки с помощью метода градиентного спуска, они комбинируют кластеризацию образцов, поиск зависимостей, то есть моделируют обобщение, порождение понятий в ходе обучения. Рассмотрим далее базовый механизм обобщения — кластеризацию, в том числе нечеткую. Приведем постановку задачи нечеткой кластеризации.

Нечеткая кластеризация

Алгоритм нечеткой кластеризации называют FCM-алгоритмом (Fuzzy Classifier Means, Fuzzy C-Means). Целью FCM-алгоритма кластеризации является автоматическая классификация множества объектов, которые задаются векторами признаков в пространстве признаков. Другими словами, такой алгоритм определяет кластеры и, соответственно, классифицирует объекты. Кластеры представляются нечеткими множествами и, кроме того, границы между кластерами также являются нечеткими. FCM-алгоритм кластеризации предполагает, что объекты принадлежат всем кластерам с определенной функцией принадлежности (ФП). Степень принадлежности определяется расстоянием от объекта

Для заданного множества K входных векторов x_k и N выделяемых кластеров c_j предполагается, что любой x_k принадлежит любому c_j со степенью принадлежности μ_{jk} , где j — номер кластера, а k — номер входного

вектора. Принимаются во внимание следующие условия нормирования для μ_{jk} :

$$\sum_{j=1}^N \mu_{jk} = 1, \text{ для всех } k = 1, \dots, K,$$

$$0 \leq \sum_{j=1}^N \mu_{jk} \leq K, \text{ для всех } j = 1, \dots, N.$$

Цель алгоритма заключается в минимизации суммы всех взвешенных расстояний $\|x_k - c_j\|$

$$\sum_{j=1}^N \sum_{k=1}^K (\mu_{jk})^q \|x_k - c_j\| \rightarrow \min,$$

где q — фиксированный параметр, задаваемый перед итерациями. Для достижения вышеуказанной цели необходимо решить следующую систему уравнений:

$$\frac{\partial \sum_{j=1}^N \sum_{k=1}^K (\mu_{jk})^q \|x_k - c_j\|}{\partial \mu_{jk}} = 0,$$

$$\frac{\partial \sum_{j=1}^N \sum_{k=1}^K (\mu_{jk})^q \|x_k - c_j\|}{\partial c_j} = 0.$$

Совместно с условиями нормирования μ_{jk} данная система дифференциальных уравнений имеет следующее решение:

$$c_j = \frac{\sum_{k=1}^K (\mu_{jk})^q x_k}{\sum_{k=1}^K (\mu_{jk})^q}$$

(взвешенный центр гравитации) и

$$\mu_{jk} = \frac{\frac{1}{\|x_k - c_j\|^{\frac{1}{(q-1)}}}}{\sum_{j=1}^N \frac{1}{\|x_k - c_j\|^{\frac{1}{(q-1)}}}}.$$

Задача исследования соотношения осознанного и подсознательного

Особенностью гибридных систем является их принципиальная интерпретируемость, то есть как и всякая система логического вывода, гибридная система объясняет свой результат с помощью обратного просмотра протокола применяемых вербализованных правил. Любая нечеткая нейронная сеть работает как система нечеткого логического вывода, но строится не с помощью инженерии знаний, а с помощью обучения по образцам. В результате матрица весов отражает силу связи входных и выходных переменных. Результатом обучения нечеткой нейронной сети служит не только матрица весов, но совокупность правил и оценок их достоверности. Следовательно, любая гибридная система является как минимум двухуровневой, включающей систему и метасистему, отражающую систему первого уровня. Исходя из сказанного, можно предложить нечеткие нейронные сети в качестве когнитивной модели соотношения сознательного (логического) и бессознательного (аналогового, вычислительного) процессов в решении интеллектуальной задачи. Таким образом, в нечеткой нейронной сети возможно сочетать манипулирование образами, заданными количественными параметрами, с преобразованием символов (слов). Такая возможность заложена в самой базовой конструкции теории нечетких множеств. Каждое нечеткое множество связывает слово, имя, с порядковой или метрической шкалой с помощью функции принадлежности, т. е. количественно моделирует новое качество — смысл.

ННС и возможности генерации правил поведения

Эффективность аппарата нейросетей определяется их аппроксимирующей способностью, причем НС являются универсальными функциональными аппроксиматорами [2]. С помощью НС можно выразить любую непрерывную функциональную зависимость на основе обучения НС, без предварительной аналитической работы по выявлению правил зависимости выхода от входа. Недостатком нейросетей является невозможность объяснить выходной результат, так как значения распределены по нейронам в виде значений коэффициентов весов. Основной трудностью в применении нечетких экспертных систем служит необходимость явно сформулировать правила проблемной области в форме продукции. В нечетких экспертных системах легко построить объяснение результата в форме протокола рас-

суждений. Поэтому в настоящее время создаются гибридные технологии, сочетающие преимущества нечетких систем и нейронных сетей. Примером гибридной технологии служит реализация системы нечетких правил на основе нейросети. База нечетких правил для двух входных и одной выходной переменных имеет следующую структуру:

R_i : **если** $(x_{1i} - A_{1i})$ **И** $(x_{2i} - A_{2i})$ **то** $(z_i - C_i)$.

Для реализации базы нечетких правил будем интерпретировать ее как таблицу определения некоторой функции, то есть базу правил можно представить обучающей выборкой: $((A_{1i}, A_{2i}), C_i)$. Например, $((\text{малое}, \text{большое}), \text{около нуля})$.

Искусственный интеллект, нейроинтеллект, естественный интеллект

Естественный интеллект сформировался в ходе многомиллиардной эволюции. Искусственный интеллект создается в научном сообществе, довольно нескоординированном, последние 40 лет. Если учесть, что человечество не знает ответа на вопрос, как работает человеческий мозг, то силы явно не равны. Не существует общепризнанного определения естественного интеллекта. Разброс мнений колеблется от «нечто, что измеряется в интеллектуальных тестах», до «способности решать задачи с неизвестным алгоритмом решения». Для настоящего момента развития ИИ характерно внимание к эволюционному процессу развития жизни, который и создал естественный интеллект [3]. Создаются системы, моделирующие эволюцию, но не живых организмов, а машин. Однако такой способ создания искусственного разума потребует геологических эпох, несмотря на быстрое действие компьютеров. От примитивной лабиринтной модели мышления ИИ быстро перешел к моделированию свойств естественного интеллекта, проявляемых в ходе филогенеза и этногенеза интеллекта. Филогенез естественного интеллекта представляет собой сложное взаимодействие развития движений и пространственных моделей окружающего мира. Телевосприятие и теледействие потребовали развития моторики, а органы движения живого представляли собой пространственную модель мира. Осуществление все более сложных движений привело к развитию систем управления, то есть различных оценочных функций, безусловных и условных рефлексов. Усложнение движений и поведения в целом привело к формированию сложных врожденных программ движения и поведения — инстинктов. Наука в современном состоянии не может ответить на вопрос, когда и как

появилась память, но ее появление позволило накапливать и использовать прижизненный опыт. Рано появившаяся асимметрия мозга привела к разделению функций, к формированию оценки (осознания) интеллектуальной деятельности и в конце концов к формированию психики и сознания. Данные онтогенеза объясняют явление интроспекции, которое заключается в том, что при освоении какой-то деятельности ребенок помогает себе движениями, которые все менее активно используются при дальнейшем росте. Например, решая лабиринтную задачу, ребенок водит пальцем, а взрослый человек «следит глазами». Приведенные данные говорят о связи движения, поведения, жизни с появлением интеллекта. Иначе говоря, чтобы машины получили интеллект в ходе эволюции, они должны начать жить.

Задачи искусственного интеллекта: Возможности ННС

Научное направление «искусственный интеллект» (ИИ)

В XIII веке средневековый испанский философ, математик, поэт *Раймонд Луллий* пытался создать механическую машину для решения различных задач на основе всеобщей классификации понятий. В XVIII веке *Лейбниц* и *Декарт* независимо предложили универсальные языки классификации всех наук. Эти работы можно считать первыми работами по теории искусственного интеллекта [4]. В 40-е годы XX века *Н. Винер* создал основополагающие работы по кибернетике. Термин «искусственный интеллект» предложен в 1956 году на семинаре в Дартмутском колледже (США). В английском языке существуют два термина: *intelligence* (умение рассуждать разумно) и *intellect* (собственно семантический аналог русского слова «интеллект»). Таким образом, английский термин изначально был лишен того фантастического семантического оттенка, которым обладает русскоязычный термин. Но и в английском языке у нового термина были свои проблемы, в частности общий корень слов *artificial* (искусственный) и *artifice* (искусственный прием, трюк), в результате ряд критиков нового направления переводил наименование «кафедра искусственного интеллекта» на вычислительном факультете как «кафедра удивительных трюков». Первые сторонники и творцы направления и сами не отказывались от слова «трюк», уточняя, что введение термина — это трюк в стиле “*Lifemanship*”, в англоязычных странах так называют всякую настойчивую инициативу, успех которой полностью зависит от демонстрации абсолютной уверенности ее

организаторов.

В истории искусственного интеллекта можно выделить термины-предшественники: кибернетика, информатика, вычислительная техника; термины-попутчики: бионика, психоника, распознавание образов, нейроинформатика и термины-потомки: инженерия знаний, вычислительный интеллект, мягкие вычисления. Почему научные направления, моделирующие интеллектуальную деятельность человека с разных точек зрения, такие как нейроматематика, теория роботов, распознавание образов, бионика, искусственный интеллект входят в информатику как отдельные направления, отличающиеся языком, методами и кадровым потенциалом? Разделение исследователей произошло из-за разных ответов на вопрос: что такое интеллект и как можно построить разумную машину? Часть исследователей отвечали на вопрос таким образом: интеллект — это способность человека к формально-логической обработке символической информации. А разумная машина должна эффективно перерабатывать символы (*Минский, Маккарти, Робинсон, Кольмероэ*). Таким образом, важен знак, символ, слово, понятие. Со временем именно данное направление стали называть «искусственный интеллект». Часть исследователей отвечали на вопрос так: интеллект — это способность человеческого мозга к целенаправленной деятельности, сложная система рефлексов, инстинктов, логической обработки информации. Думающая машина может быть только подобна естественному интеллекту (*Розенблатт, МакКаллок, Хебб*). Таким образом, важен сигнал, ощущение, восприятие. Сторонники направления работают в направлениях — нейроинформатика, распознавание образов, бионика. В ходе формирования науки искусственного интеллекта, определения ее содержания, структуры, определялся и специфический предмет исследования. Задача была и остается сложной, так как наука обладает принципиальной междисциплинарностью. Сложность ее истории в значительной степени определяется мировоззренческим потенциалом искусственного интеллекта.

Состояние ИИ

К настоящему времени предмет искусственного интеллекта остался по-прежнему неопределенным, что не мешает конкретным исследованиям быть полезными. Перечислим только основные достижения ИИ:

- создание систем, основанных на знаниях (knowledge based system), решение задач представления, обработки и извлечения знаний, разработка многочисленных экспертных систем;

- разработка естественно-языковых интерфейсов и развитие машинного перевода. Хотя ни одна из этих задач и не решена окончательно, но продвижение в указанных направлениях существенное;
- развитие математической логики — от метода резолюций *Робинсона* до развития нетрадиционных логик (немонотонных, правдоподобного вывода и т. д.);

Значительное влияние ИИ оказал на развитие индустрии программирования. В частности, язык ЛИСП был предложен *Маккарти* для решения задач именно ИИ, а сейчас используется более широко. Парадигма объектно-ориентированного программирования сложилась в ходе проведения работ по ИИ. Анализ показывает сближение представителей разных направлений ИИ: традиционная обработка знаний и естественные языки; нейронные сети; распознавание и синтез речи; многоагентные системы. Сближение связано не только с естественными циклами развития любой науки, но и с тенденцией глубинной интеграции. На наш взгляд важнейшая тенденция в развитии ИИ ближайшего будущего — это интеграция автономных достижений, причем глубинная интеграция, осуществляемая на основе создания гибридных систем. Трудно сказать под каким именем будет происходить такая широкая гибридизация, возможна регенерация термина «гибридные системы» или развитие термина «синергетический интеллект». В любом случае впереди гибридизация достижений инженерии знаний, распознавания образов, нейроинформатики, роботики, которая должна обеспечить синергизм сигнала, числа, слова и понятия. В самом общем виде такую тенденцию можно выразить как переход от задачи создания думающей машины к задаче создания интеллектуального существа.

Искусственный и вычислительный интеллект

К. Циммерман (Аахен, Германия) предложил в начале 90-х годов термин «вычислительный интеллект». Синонимичный «зонтичный» термин предложил в 1994 году на семинаре в Беркли *Л. Заде* — «мягкие вычисления», включающий нечеткую логику, нейронные вычисления и эволюционные вычисления. Рождение новых направлений, а, значит, и развитие науки продолжается.

Традиционная задача ИИ — инженерия знаний и экспертные системы

Экспертные системы (ЭС) — это прикладные интеллектуальные системы, в которых база знаний представляет собой формализованные эмпирические знания высококвалифицированных специалистов (экспертов) в какой-либо узкой предметной области.

Экспертная система состоит из следующих компонент:

- *База знаний* предназначена для хранения экспертных знаний о предметной области, используемых при решении задач экспертной системой. База знаний состоит из набора фреймов и правил-продукций.
 - Фреймы используются в базе знаний для описания объектов, событий, ситуаций, прочих понятий и взаимосвязей между ними. Фрейм — это структура данных, состоящая из слотов (полей).
 - Правила используются в базе знаний для описания отношений между объектами, событиями, ситуациями и прочими понятиями. На основе отношений, задаваемых в правилах, выполняется логический вывод. В условиях и заключениях правил присутствуют ссылки на фреймы и их слоты.
- *База данных* предназначена для временного хранения фактов или гипотез, являющихся промежуточными решениями или результатом общения системы с внешней средой, в качестве которой обычно выступает человек, ведущий диалог.
- *Машина логического вывода* — механизм рассуждений, оперирующий знаниями и данными с целью получения новых данных из знаний и других данных, имеющихся в рабочей памяти. Для этого обычно используется программно реализованный механизм дедуктивного логического вывода (какая-либо его разновидность) или механизм поиска решения в сети фреймов или семантической сети. Машина логического вывода может реализовывать рассуждения в виде:
 - дедуктивного вывода (прямого, обратного, смешанного);
 - нечеткого вывода;
 - вероятностного вывода;
 - унификации (подобно тому, как это реализовано в Прологе);
 - поиска решения с разбиением на последовательность подзадач;
 - поиска решения с использованием стратегии разбиения пространства поиска с учетом уровней абстрагирования решения или понятий, с ними связанных;

- монотонного или немонотонного рассуждения,
 - рассуждений с использованием механизма аргументации;
 - ассоциативного поиска с использованием нейронных сетей;
 - вывода с использованием механизма лингвистической переменной.
- *Подсистема общения* служит для ведения диалога с пользователем, в ходе которого ЭС запрашивает у пользователя необходимые факты для процесса рассуждения, а также, дающая возможность пользователю в какой-то степени контролировать и корректировать ход рассуждений экспертной системы.
 - *Подсистема объяснений* необходима для того, чтобы дать возможность пользователю контролировать ход рассуждений и, может быть, учиться у экспертной системы.
 - *Подсистема приобретения знаний* служит для корректировки и пополнения базы знаний. В простейшем случае это — интеллектуальный редактор базы знаний, в более сложных экспертных системах — средства для извлечения знаний из баз данных, неструктурированного текста, графической информации и т. д.

Нечеткая ЭС — экспертная система, которая для вывода решения использует совокупность нечетких функций принадлежности и правил. Функция принадлежности выражает степень соответствия элемента некоторому понятию. Реальным источником ее формирования являются расспросы эксперта и обобщение статистических данных.

Мягкие экспертные системы

Рассмотрим архитектуру и основные структурно-функциональные решения мягкой экспертной системы (МЭС). Для определения МЭС, сопоставим понятия нечеткой и мягкой экспертных систем. Для описания архитектуры МЭС будем использовать три признака:

- способ извлечения знаний;
- представление знаний;
- обработка знаний.

Перечисленные признаки создают общую «координатную сетку» описания.

Сравнение нечеткой и мягкой экспертной систем. Нечеткие экспертные системы используют представление знаний в форме нечетких продукций и лингвистических переменных. Основу представления лингвистической переменной составляет терм с функцией принадлежности. Способ обработки знаний в нечетких ЭС — это логический вывод по нечетким продукциям. Особенностью нечеткой ЭС является способ извлечения функций принадлежности, который сводится либо к статистическим методам построения, либо к методу экспертных оценок. Мягкой ЭС будем называть нечеткую ЭС, которая обладает следующими особенностями:

- Мягкая ЭС для извлечения знаний использует статистические данные, которые интерпретирует как обучающие выборки для нечетких нейронных сетей.
- Мягкая ЭС представляет знания в виде лингвистических переменных (функций принадлежности), нечетких продукций и обученных нейронных сетей. Редукция множества нечетких продукций, настройка ФП и базы правил выполняется с помощью ГА.

Мягкими называют вычисления, сочетающие теорию нечетких систем, нейронные сети, вероятностные рассуждения и генетические алгоритмы и обладающие синергическим эффектом, следовательно, мягкой ЭС называют ЭС, сочетающую перечисленные теории ради того же эффекта взаимного усиления.

Инженерия знаний и мягкие экспертные системы. Мягкая ЭС извлекает знания из статистических выборок, интерпретируя их как обучающие выборки для обучения нейронной сети. В дальнейшем особенности мягкого извлечения знаний продемонстрируем на двух примерах:

- извлечение нечетких продукций для ЭС анализа тенденций развития предприятия;
- извлечение нечетких продукций для задачи конструирования стенов изделий.

Представление знаний в мягкой экспертной системе. Содержание баз знаний и данных мягкой экспертной системы. Если использовать ННС на этапе извлечения знаний, то кроме функций принадлежности и нечетких продукций, порождается совокупность обученных НС, которые входят в базу знаний МЭС. Оптимизация (редукция) множества извлеченных правил выполняется на основе генетического алгоритма.

База знаний МЭС должна содержать следующие части:

- функции принадлежности;
- нечеткие продукции;
- обученные нечеткие нейронные сети;
- процедуры интерпретации хромосом генетических алгоритмов;
- функции оптимальности.

Рассмотрим проблему представления перечисленных составных частей в компьютерных интеллектуальных системах. Если функция принадлежности характеризуется такими математическими свойствами, как непрерывность, выпуклость (унимодалность), то функция принадлежности может быть представлена параметризованной функцией формы. Наибольшее распространение получили три вида функций формы: треугольная, трапециевидная и колоколообразная (сигмоидальная); которые определяются тройкой, четверкой и двойкой параметров соответственно.

Пример мягкой ЭС для проектирования радиоэлектронных изделий

Описание задачи конструирования РЭА. Для производственной деятельности конструкторских бюро характерно изготовление опытных образцов, малых серий изделий, модифицированных блоков серийных изделий. Задачей МЭС служит разработка стендов для испытаний или контроля и диагностики блоков. Большие конструкторские бюро, имеющие длительную историю создания таких стендов обладают значительными архивами разнообразных стендов. Архивы проектов в настоящее время представляют собой хранилища бумажной технической документации. Информация, хранящаяся таким образом, редко используется повторно и достаточно оперативно. Другим результатом продолжительной производственной деятельности конструкторских бюро является наличие опытных специалистов-конструкторов. Такие специалисты, эксперты, покидая предприятие часто уносят уникальный опыт проектирования. Поэтому в качестве решения проблемы сохранения накопленного опыта проектирования необходимо комплексное решение. Недостаточно использовать даже самую современную распределенную информационную систему доступа к архиву проектов, так как методика конструирования не отражается в конечном продукте — технической документации на стенд. Информационная система должна включать в себя экспертную систему, содержащую знания экспертов,

Таблица 1. Средства решения задач конструирования изделий радио-электронной аппаратуры (РЭА)

Задача	Средство решения
Выбор корпуса	Система нечеткого вывода
Выбор типа корпуса	Система принятия решений
Размещение элементов	Генетический алгоритм

которые в настоящее время никак не документируются на предприятиях. Для создания в качестве интеллектуальной компоненты именно экспертной системы имеются серьезные предпосылки: наличие экспертов; наличие архива проектов из которого можно извлекать знания, возможность участвовать в конструировании какого-либо стенда, разрабатываемого в проектной организации. Для конструирования стендов выберем средства решения, результаты выбора приведены в табл. 1.

Нейросетевая реализация МЭС конструирования стендов контроля РЭА. Учитывая значительный объем реальных архивов проектных организаций, особенно привлекательным представляется извлечение знаний из данных для организации процесса извлечения знаний необходимо выделить факторы влияния в качестве базовых переменных и составить таблицы значений этих переменных для проектов, хранимых в архиве. Выходная переменная — это переменная, обозначающая тип и размер корпуса. В качестве базовых переменных были выделены следующие:

- наличие элементов на передней панели с рабочим положением «Вертикальное»;
- наличие индикаторов на передней панели с небольшим углом обзора;
- количество регулируемых элементов;
- количество ячеек;
- суммарная площадь ячеек;
- количество элементов, устанавливаемых внутри блока;
- суммарный объем элементов, устанавливаемых внутри блока;
- суммарная установочная площадь внутри блока;
- наличие элементов внутри блока с креплением «шасси»;
- количество тумблеров на передней панели;
- количество элементов со шкалами;

- количество стрелочных индикаторов;
- количество галетных переключателей;
- суммарная установочная площадь на передней панели.

Для извлечения правил необходима нейронная нечеткая сеть, вербализующая правила. Таким инструментом может служить ННС с редукцией полученных правил генетическим алгоритмом FUNGEN 1.1. Обучающая выборка должна представлять собой множество строк, каждая строка содержит сведения об одном стенде. Структура строки содержит две части: входные переменные, определяющие выбор корпуса или передней панели и сделанный выбор — тип и габариты корпуса или размеры передней панели. В результате обучения ННС не только отбираются правила вывода, но и настраиваются параметры функций принадлежности. Для извлечения правил необходима нейронная нечеткая сеть, вербализующая правила. Таким инструментом может служить ННС с редукцией полученных правил генетическим алгоритмом FUNGEN 1.1. Обучающая выборка должна представлять собой множество строк, каждая строка содержит сведения об одном стенде. Структура строки содержит две части: входные переменные, определяющие выбор корпуса или передней панели и сделанный выбор — тип и габариты корпуса или размеры передней панели. В результате обучения ННС не только отбираются правила вывода, но и настраиваются параметры функций принадлежности. Уточненные функции принадлежности являются результатами эксперимента. Для каждой входной и выходной переменных разработчик МЭС составляет первоначальные функции принадлежности нечетких термов. Правила, полученные в результате обучения ННС выражают аналогию между стендами, т.е. для стендов с похожими структурно-функциональными элементами рекомендуются одинаковые корпуса, а ННС оценивает их близость. Исходные лингвистические переменные и их нечеткие метки определены на основе экспертных данных и анализа истории проектирования стендов изделий. На основе выделенных с помощью классификаций базовых переменных и задания их семантики с помощью функций принадлежности можно сформулировать правило вывода проектных решений. Правила конструирования стендов определяют степень возможности выбора типа и типоразмера блока с точки зрения элементной базы и составляют основную часть экспертной системы. Левые части правил определяют нечеткие значения значимых лингвистических переменных, а правые определяют степень возможности выбора каждого из допустимых типов корпуса.

Задачи прикладного ИИ: Возможности ННС

Интеллектуальный анализ данных

В настоящее время накоплены огромные объемы информации, описывающие различные сложные системы. Анализ этих данных, содержащих свойства объектов, позволяет описать реальность и построить модель для дальнейшего изучения и прогнозирования поведения системы. Результатом анализа является нахождение взаимосвязей, динамики развития свойств объектов, т.е. знание о предметной области. Знание может выражаться функциональными зависимостями между свойствами, логическими связями. Исследования данных и их методов анализа оформились в виде отдельного направления называемого Data Mining (DM). Современные комплексы DM позволяют «добывать» знания о системах в виде простых или сложных функций, множества продукции {если... то...} и представляют собой инструменты интеллектуального анализа данных. Каждый метод DM имеет свои достоинства и недостатки, что обусловлено характером данных и их взаимосвязей. Один из видов извлекаемых знаний — это знания о динамике развития объекта. Существует статистическое решение задачи выделения тренда — постоянной составляющей временного ряда. Аналитическое выражение тренда можно найти методами регрессионного анализа, сезонной декомпозиции, анализа Фурье, различными методами сглаживания. К их общим недостаткам следует отнести необходимость задания явной параметрической модели тренда и требования стационарности остатка. Подход с точки зрения вычислительного интеллекта (мягких вычислений) позволяет использовать для распознавания тенденций аппарат нейронных сетей. Социологические, экономические, экологические базы данных (БД) содержат лингвистические оценки и являются нечеткими реляционными БД. Поэтому извлечение знаний (Data Mining) предполагает решение двух задач:

- кластеризации;
- поиска зависимостей.

ННС как Data Miner для нечетких реляционных баз данных

Нечеткая реляционная модель данных. Способ представления данных. Нечетким отношением будем называть конечное множество отображений, содержащих нечеткие данные. Нечеткие данные могут присутство-

вать только на определенных атрибутах — нечетких атрибутах. Нечеткий атрибут определим как атрибут, на домене которого можно определить нечеткое множество. Модель рассчитана на представление нечетких чисел, соответственно доменом атрибута нечеткого числа является множество действительных чисел. Нечеткое число может быть задано в виде функции принадлежности (лингвистической оценки). Под лингвистической оценкой будем понимать одно из возможных значений лингвистической переменной, которое определяется соответствующим термом. Функция принадлежности, задающая нечеткое число удовлетворяет следующим свойствам:

- ограниченности (по определению $\max(\mu(x)) \leq 1, \min(\mu(x)) \geq 0$);
- однозначности (каждое множество $\mu(x) = Z, x \in X$, состоит только из одного элемента);
- непрерывности (в каждой точке существует предел функции).

Если для некоторого аргумента значение функции принадлежности неопределенно, то предполагается, что это значение равно нулю. Для представления точных (четких) значений используется вырожденный вид функции принадлежности, возвращающей 1 для представляемого точного значения и 0 для всех остальных значений.

В модели принят табличный способ задания функция принадлежности. Количество пар $\mu(x_i)/x_i$, задающих функцию принадлежности, не ограничено. Предполагается, что множество таких пар задает ломаную линию, которая и является графиком функции принадлежности [6].

Кластеризация реляционных кортежей. Задача кластеризации заключается в разбиении объектов на кластеры, когда основой разбиения служит вектор параметров объекта. Объекты в пределах одного класса считаются эквивалентными с точки зрения критерия разбиения. Сами кластеры часто бывают неизвестны заранее, а формируются динамически. Кластеры зависят от предъявляемых объектов, и поэтому добавление нового объекта требует корректирования системы кластеров.

Будем характеризовать объекты, подлежащие кластеризации, вектором $x^p \in X$, имеющим N компонент, компоненты обозначим нижним индексом: $x^p = (x_1^p, \dots, x_N^p)^T$. Вектор параметров — единственная характеристика объектов при их кластеризации. Введем множество кластеров $C^1, \dots, C^M = C^m$ в пространстве кластеров C :

$$(C^1 \cup C^2 \dots \cup C^M) \subset C.$$

Пространство кластеров может не совпадать с пространством объектов X и иметь другую размерность. В простейшем случае, когда пространства кластеров и объектов совпадают, кластеры представляют собой области пространства X , и объект x^p будет отнесен к одному из классов m_0 , если $x^p \in C$. В общем случае X и C различны.

Определим ядра кластеров $C = c^1, \dots, c^m$ в пространстве классов C , как объекты, типичские для своего класса. Очевидно, что близость объекта к ядру необходимо оценивать численно. Введем меру близости $d(x^p, c^m)$ — скалярную функцию от объекта и ядра кластера, которая тем меньше, чем больше объект похож на ядро кластера. Могут потребоваться вспомогательные меры близости, определенные для двух объектов, $d(x^p, c^m)$, и для двух ядер классов, $d(c^{m_1}, c^{m_2})$. Чаще всего применяется евклидова мера:

$$d(x, y) = \sum_i (x_i - y_i)^2$$

или

$$d(x, y) = \sum_i \|(x_i - y_i)\|.$$

Задавшись числом кластеров, можно поставить задачу кластеризации: найти M ядер кластеров C^m и разбить объекты X^M на кластеры C^m , то есть построить функцию $m(p)$ таким образом, чтобы минимизировать сумму мер близости:

$$m(D) = \sum_p (d(x^p, c^{m(p)})).$$

Функция $m(p)$, определяющая номер кластера по индексу p множества объектов X^p , задает разбиение на кластеры и является решением задачи кластеризации. В простейшем случае $X = C$, пространство объектов X разбивается на области C^m и если $x_0^p \in C_0^m$, то $(p_0) = m_0$, тогда объект относят к кластеру m_0 . Количество кластеров может динамически меняться. При этом часто возникают ситуации, когда объекты распределены по кластерам неравномерно. Необходимо контролировать равномерность плотности ядер c^m в пространстве C и долю объектов, относящихся к каждому кластеру m_0 . Рассмотрим общий алгоритм с фиксированным количеством ядер M . Количество ядер выбирается заранее, исходя из конкретной задачи. Начальные значения ядер могут выбираться случайными c^1, \dots, c^m , одинаковыми или по другим эвристическим правилам. Каждая итерация алгоритма состоит из двух этапов:

- При неизменных ядрах $C^m = \text{const}$ ищем такое разбиение $m(p)$ объектов X^p на классы, чтобы минимизировать суммарную меру близости между объектами и ядрами их кластеров:

$$\min \left[\sum_p (d(x^p, c^{m(p)})) \right].$$

Результат этапа — создание функции $m(p)$, разбивающей объекты на кластеры.

- При измененном разбиении $m(p)$ настраиваем ядра C^m так, чтобы в пределах каждого кластера m_0 суммарная мера близости ядра этого кластера и объектов, ему принадлежащих, была минимальной: $\sum_p d[x^p, c^{m(p)}]$ для всех $m_0 = 1, \dots, M$.

Результат этого этапа — новый набор ядер C^m .

Решение задачи нечеткой кластеризации на основе сети Кохонена.

Выберем в качестве входных данных вектор параметров единственного объекта. Результатом работы сети будет код кластера, к которому принадлежит предъявленный на входе объект. Сеть будет иметь M выходов, по числу кластеров, и чем большее значение принимает выход номер m_0 , тем больше «уверенность» сети в том, что входной объект принадлежит к кластеру m_0 . Если применить функцию активации SOFTMAX, тогда сумма выходов всегда будет равна единице. Каждый выход можно будет трактовать как степень уверенности того, что объект принадлежит данному кластеру. Выберем евклидову меру близости. В этом случае ядро кластера, минимизирующее сумму мер близости для объектов этого кластера, совпадает с центром тяжести объектов:

$$c^{m_0} = \frac{1}{N(m_0)} \sum_{p:m(p)=m_0} x^p,$$

где $N(m_0)$ — число объектов x^p в кластере m_0 . При разбиении на кластеры должна быть минимизирована суммарная мера близости для всего множества X^p входных объектов:

$$D = \sum_p \sum_i [x_i^p - c_i^{m(p)}]^2 = \sum_p [(x^p, x^p) - 2(x^p, c^{m(p)}) + (c^{m(p)}, c^{m(p)})]$$

расписано скалярное произведение. В этой сумме два слагаемых не зависят от способа разбиения и постоянны:

$$\sum_p (c^{m(p)}, c^{m(p)}) = \text{const},$$

$$\sum_p (x^p, x^p) = \text{const}.$$

Поэтому задача поиска минимума D эквивалентна поиску максимума выражения:

$$\min D \rightarrow \max \sum_p \sum_i x_i^p c_i^{m(p)}.$$

Вариант алгоритма кластеризации для поиска максимума этой функции:

Алгоритм 1: Алгоритм кластеризации

```

for all Для каждого вектора  $x^p$  do
  for all Для каждого  $m$  do
    Рассчитать  $\sum_i x_i^p c_i^m = D^{m,p}$ 
  end for
  Найти  $m_0$ , для которого  $m_0 : \max(D^{m,p})$ 
  Отнести объект к кластеру  $m_0$ 
end for

```

Данный алгоритм легко реализуется в виде нейронной сети. Для этого требуется M сумматоров, находящих все $D^{m,p}$, а также интерпретатор, находящий сумматор с максимальным выходом. Входные векторы сети чаще всего нормируются:

$$\frac{x^p}{\|x^p\|}; \quad \frac{x^p}{\sum_p \|x^p\|^2} \rightarrow x^p; \quad \frac{x}{\sqrt{\sum_p \|x^p\|^2}} \rightarrow x^p.$$

Возможны и другие способы нормировки.

Обучение сети. Задача обучения — научить сеть активировать один и тот же нейрон для похожих векторов x^p на входе. Не важно, какой конкретно нейрон будет активирован. Обычно начальные значения в нейронных

сетях выбираются малыми случайными числами. В случае, если ядра кластеров нормированы, то и начальные значения необходимо нормировать. Если веса инициализируются случайными значениями с равномерным распределением, то возникает проблема. Когда ядра распределяются равномерно, то в областях пространства, где мало входных векторов, ядра будут использоваться редко, так как мало будет похожих векторов. В тех областях, где входных векторов много, плотность ядер окажется недостаточной, и непохожие объекты будут активировать один и тот же нейрон, так как более похожего ядра не найдется. Для устранения проблемы можно выделять ядра в соответствии с плотностью входных векторов. Но распределение входных векторов часто бывает заранее неизвестно. В этом случае возможно использование метода выпуклой комбинации. Если число входных векторов равно числу ядер, то обучение не нужно. Достаточно присвоить ядрам значения входных векторов, и каждый вектор будет активировать свой нейрон Кохонена. Но чаще всего количество кластеров меньше числа входных векторов. В этом случае веса сети настраиваются итеративным алгоритмом.

Алгоритм аналогичен исходному алгоритму классификации, но коррекции весов проводятся после предъявления каждого входного вектора, а не после предъявления всех, как требует исходный алгоритм. Сходимость при этом сохраняется.

1. Присваиваем начальные значения весовым коэффициентам.
2. Подаем на вход один из векторов x^p .
3. Рассчитываем выход слоя Кохонена, $D^{m,p}$, после чего определяем номер выигравшего нейрона m_0 , выход которого максимален, $m_0 : \max D^{m,p}$.
4. Корректируем веса только выигравшего нейрона m_0 :

$$\omega_m := \omega_m + \alpha(x^p - \omega_m);$$

коррекция записана в виде векторного выражения (вектор весов ω_{m_0} нейрона m_0 имеет столько компонент, сколько их у входного вектора x^p). Здесь α — скорость обучения, малая положительная величина. Часто используют расписание с обучением, когда $\alpha = \alpha(t)$ монотонно убывает.

Веса корректируются так, что вектор весов приближается к текущему входному вектору. Скорость обучения управляет быстротой приближения ядра

кластера (вектора весов) ко входному вектору x^p . Алгоритм выполняется до тех пор, пока веса не перестанут меняться.

Режимы работы сети. Обычная сеть Кохонена работает в режиме аккредитации. Это означает, что активируется единственный нейрон Кохонена с максимальным значением выхода. Можно не затормаживать остальные нейроны слоя Кохонена, а пронормировать выходные сигналы, например, функцией SOFTMAX:

$$OUT_j = \frac{e^{NET_j}}{\sum_i e^{NET_i}}.$$

Тогда сумма всех выходов слоя будет равна единице и можно трактовать выходы как степень принадлежности объекта к каждому из классов. Такой режим работы сети, когда активируется несколько нейронов одновременно, называется режимом интерполяции. Название режима объясняется тем, что если входной вектор $x^p = \omega^{m_1}$ плавно меняется от одного вектора весов $x^p = \omega^{m_1}$ к другому вектору весов $x^p = \omega^{m_2}$, то выход сети в режиме интерполяции (если применена функция SOFTMAX) будет плавно меняться от m_1 к m_2 , то есть классификация оказывается непрерывной. Если же сеть работает в режиме аккредитации, выход изменяется от m_1 к m_2 скачкообразно.

Извлечение правил из реляционных кортежей с помощью ННС

Прогнозирование нечетких временных рядов. Подход с точки зрения мягких вычислений позволит использовать для распознавания тенденций аппарат нейронных сетей. Исходные данные могут быть представлены в нечетком выражении, что может быть обусловлено экспертным оцениванием состояния системы, а так же несовершенными способами измерения параметров. Для описания изменений этих нечетких величин вводится понятие нечеткого ряда: упорядоченная последовательность нечетких меток, другими словами, нечеткая зависимость от аргумента. Наиболее часто изменения параметра рассматриваются во времени, в этом случае говорится о нечетком временном ряде, где время есть свойство системы. Тенденция изменений так же выражается нечеткостью, являясь «нечеткостью второго порядка». Носителем нечеткого множества тенденций является множество всевозможных функций.

В простейшем случае тенденция может описываться лингвистическими терминами «рост», «стабилизация», «падение», оценивающие знак производной функции. Но под нечеткой тенденцией можно понимать образы любых функции, не только «рост», «падение», «стабилизация», но и экспертные оценки тенденций: «колебания», «хаос». Таким образом, текущие состояние системы можно описать нечеткими тенденциями ее параметров.

Определение нечеткого временного ряда Нечетким временным рядом (НВР) будем называть упорядоченную последовательность наблюдений над некоторым явлением, характер которого изменяется во времени, если значения, которые принимает некоторая величина в момент времени, выражена с помощью нечеткой метки. Нечеткая метка представлена лингвистическим термом и функцией принадлежности.

Определение нечеткой тенденции. Тенденцией нечеткого временного ряда (НТ) будем называть нечеткую метку, выражающую динамику (систематическое движение) НВР. К значимым наименованиям тенденций относятся следующие нечеткие метки:

- рост;
- падение;
- стабилизация;
- колебания;
- хаос.

Разумеется, для нечетких термов, обозначающих тенденцию, возможно применение модификаторов «очень», «более-менее» и т. д. Подход с точки зрения мягких вычислений позволяет использовать для распознавания тенденций аппарат нейронных сетей. Временные ряды представляются своими графиками. Распознавание тенденции каждого графика можно считать задачей распознавания изображений. В результате подход мягких вычислений позволяет сформулировать следующее утверждение.

Результативным средством распознавания тенденций НВР является нейронная сеть. Кроме распознавания тенденций НВР рядов интерес представляет возможность извлечения правил распознавания тенденций методом нейрокомпьютинга. Нечеткое понятие «тенденция» имеет лингвистические термы «рост-развитие», «снижение-деградация», «стабилизация», «хаос».

Например, для оценки темпов роста оборота предприятия используются следующие базовые переменные: «Оборот предприятия — растет, падает, стабилизация, хаос; темпы роста — высокие, средние, низкие».

Анализ тенденций — это удобная задача для извлечения продукций, так как можно разработать генераторы тестов различных тенденций. При разработке таких генераторов необходимо учитывать диапазоны значений конкретной проблемной области, типичные значения таких переменных.

Сравним эффективность нечеткой нейронной сети архитектуры FUN с классическими НС при решении данной задачи. Для успешного обучения важна методика обучения тенденциям, которая состоит в том, что более качественные результаты распознавания дает нейронная сеть, обученная отличать один вид тенденции от всех других. Результативной является методика обучения отдельной ННС одной тенденции. Сначала — тенденции «рост», затем — «падение», «стабилизация» и другим. На заключительном этапе методика требует синтезировать нейро-нечеткую ЭС из обученных отдельных ННС, рассматриваемых как элементы. Архитектура FUN удобна для извлечения знаний, так как она позволяет выполнить настройку функций принадлежности, представленных параметризованным семейством сигмоидальных функций.

Синергические средства, такие как нечеткие нейросети с генетической редукцией правил позволяют сформулировать систему продукций для анализа тенденций на основе обучения. В качестве нейроимитатора можно выбрать нейроимитатор FUNEGEN 1.1, свободно доступный для исследователей в Internet. FUNEGEN 1.1 позволяет сформулировать систему продукций для анализа тенденций на основе обучения. Интерес представляет сравнение качества распознавания тенденций системой продукций, сформулированной экспертом, с системой продукций, сгенерированной нечетким нейроимитатором. Краткое описание нейроимитатора позволяет сделать вывод о наличии эффекта синергизма при использовании этого инструмента.

Ниже приведены сгенерированные правила распознавания.

3.892	IF (i3 mf0) THEN	output0
2.555	IF (i3 mf2) THEN	output1
1.985	IF (i0 mf3)	
	OR (i1 mf2) THEN	output2
–4.889	IF (i2 mf3) THEN	<i>NOT</i> output0
–4.565	IF (i3 mf0) THEN	<i>NOT</i> output1
–1.530	IF (i2 mf0) THEN	<i>NOT</i> output2
1.316	IF (i0 mf1)	
	OR (i3 mf2) THEN	output1
–0.909	IF (i3 mf2) THEN	<i>NOT</i> output2
0.532	IF (i0 mf0) THEN	output0
–0.511	IF (i0 mf1) THEN	<i>NOT</i> output2

Многослойная НС (11 входов и 3 выхода) представляет собой трехслойную сеть с семью нейронами в скрытом слое, 110 входами и тремя выходами и сигмоидальной функцией активации. Потребовалось 2830 итераций для уменьшения ошибки НС до уровня 10^{-4} , что обеспечивает стабильное распознавание образов на уровне 94%. Эксперимент, проводимый на сети Кохонена, проводился на той же самой обучающей выборке, что и эксперимент, проводимый на многослойной сети. Сеть Кохонена представляла собой двухслойную сеть со 110 входами и тремя выходами. В качестве функции активации нейронов использовалась униполярная сигмоида. Потребовалось 10000 итераций для стабилизации выходов с точностью в 10^{-6} на всех образцах, что обеспечивает стабильное распознавание тенденций на уровне 89%.

Результаты сравнения результативности многослойной нейронной сети и сети Кохонена для решения задачи распознавания тенденций приведены в табл. 2.

Переменные означают: X_1 — количество распознаваемых тенденций; X_2 — количество итераций для многослойной НС; X_3 — количество итераций для сети Кохонена; X_4 — процент корректного распознавания для многослойной НС; X_5 — процент корректного распознавания для сети Кохонена. Из табл. 2 видно, что сеть Кохонена обучается быстрее только при двух выходах, а при большем количестве выходов уступает в скорости обучения многослойной нейронной сети. По качеству обучения многослойная нейронная сеть предпочтительнее. Сеть Кохонена не только обучается указанным тенденциям, но выделяет классы тенденций, так как работает в режиме классификатора.

ТАБЛИЦА 2. Количество итераций обучения и процент корректного распознавания

X_1	X_2	X_3	X_4	X_5
2	2500	3300	99	99
3	3300	5000	94	89
4	9000	10000	94	88
5	15000	17500	94	87
6	20000	30000	93	86
7	23300	32000	92	86
8	25000	35000	92	85
9	30000	40000	91	85
10	35000	50000	89	85

Для этапа интегральной оценки состояния объекта можно использовать нейросеть многослойной архитектуры. Система таких нейросетей, которые называют макронейроном, способна выполнить прогнозирование в широком смысле. Ранее рассматривались аспекты анализа отдельных базовых переменных объекта, но объект экспертизы является многомерным и, разобравшись с оценками и тенденциями отдельных величин, необходимо разобраться с состоянием объекта в целом. Задача распадается на две подзадачи: выделения множества базовых переменных, зависимых переменных и зависимостей между ними, определяющими правила поведения объектов; генерации интегральной оценки состояния объекта. Первая из подзадач решается традиционными методами регрессионного анализа, вторая представляется задачей, для которой результативно использование мягких вычислений (МВ).

Прогнозирование будущего функционирования системы основано на тенденциях базовых показателей и методах аппроксимации. Для прогнозирования значений нечетких ВР можно использовать НС в режиме предиктора. НС обучается распознавать тенденции с приемлемым качеством для одного или нескольких ВР. Для получения интегральной оценки необходима совокупность НС, обученных поведению различных базовых переменных. Экспертную систему, сочетающую традиционный механизм продукционного вывода с использованием обученных НС будем называть нейроэкспертной системой. Задача прогнозирования в широком смысле, как

прогнозирование интегрального состояния объекта проектирования может быть выполнена нейроэкспертной системой.

Распознавание нечеткой тенденции (НТ) с помощью ННС. Нахождение НТ — основополагающая задача в построении системы анализа данных, решение которой позволит приступить к выявлению связей между параметрами исследуемого объекта. НТ должна обнаруживаться независимо от выбранного периода, что позволит для каждого вида тенденции разбивать исходные данные на периоды наблюдения этой НТ. Например, для текущей точки (момента времени) определяются следующие тенденции: период δt_3 — падение параметра, δt_2 — периодические колебания, а δt_1 — рост (рис. 5).

Исходные данные преобразуются в множество, которое можно представить в виде плоскости в трехмерном пространстве $(t - x - w)$, где t — время, x — исследуемый параметр, w — степень уверенности (рис. 5).

Срез по оси t есть нечеткая метка со своей функцией принадлежности. В качестве инструмента определения НТ может выступать многослойная нейронная сеть, входом которой является множество нечетких меток, а выходом значения тенденций (0 или 1). Количество выходов может варьироваться в зависимости от выбранного варианта: одна нейронная сеть (НС) — один тип НТ, одна НС — на все типы. Задача эксперта не только описать тенденцию, но также и обучить НС. Для обучения НС пользователем подготавливаются выборка данных, представляющая собой ряды нечетких меток и соответствующие им тенденции. Каждая нечеткая метка выражается вероятностью лингвистических термов: «низкое», «среднее», «высокое», т. е. тройкой чисел от 0 до 1. Например, тенденцию «период» можно представить следующим рядом чисел: «высокое» (0, 0, 1), «среднее» (0, 1, 0), «низкое» (1, 0, 0), «среднее» (0, 1, 0), «высокое» (0, 0, 1), «среднее» (0, 1, 0) и т. д. Следует отметить такую характеристику тенденции, как период ее протекания, выражающий глубину просмотра данных для текущей точки. Рассмотрим результат анализа нечетких тенденций временного ряда. Временной ряд представляет выборку недельных доходов в течение четырех лет. Для анализа определены нечеткие тенденции: рост доходов, падение доходов, стабильность, периодические колебания. Обучение нейронных сетей проводилось на основе следующих описаний эталонных тенденций: лингвистические термы нечетких меток — *низкий*, *средний*, *высокий*; период обучающей тенденции равен 7. В качестве размерности матрицы оце-

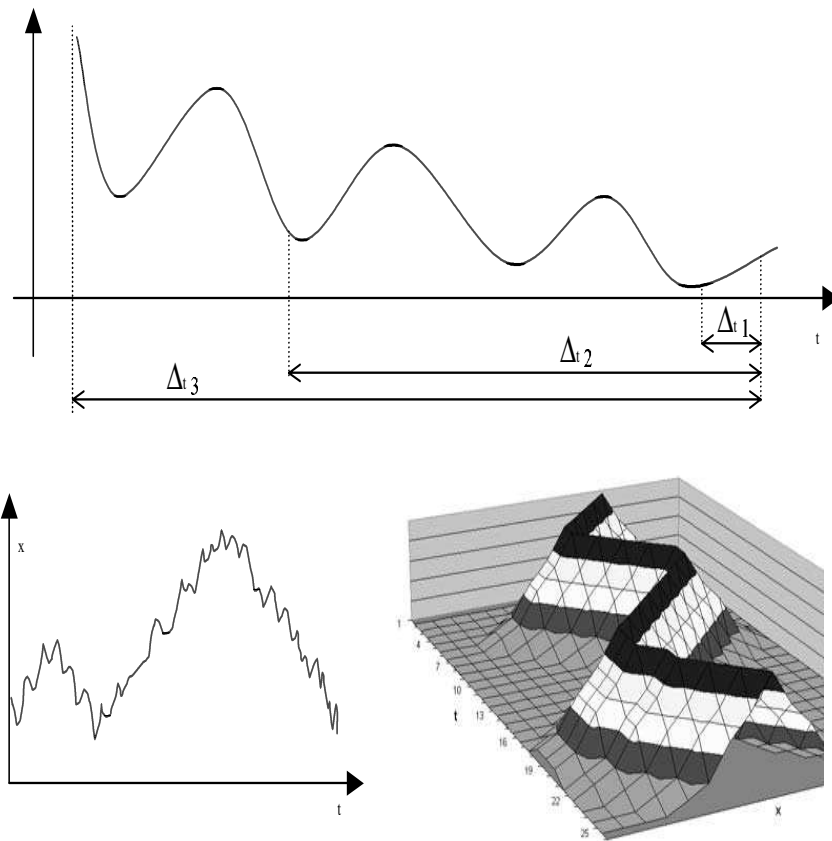


Рис. 5. Исходные данные и срез по оси t

нок выбраны следующие значения: временной ряд на моментах от 7 до 217, интервалы нечетких тенденций от 7 до 60, что соответствует поиску тенденций на интервалах от 1.5 месяца до 1.5 года. Рассмотрим результат анализа полученной матрицы оценок тенденции роста.

На рис. 6 представлены исходные данные и матрица оценки нечеткой тенденции роста доходов. Выявлены основные группы динамики роста: долгосрочный рост в интервале 9 месяцев – 1.5 года, и краткосрочный рост доходов – 2 месяца.

Нейросетевое решение задачи прикладного ИИ – развитие информационных хранилищ

Определение информационного хранилища (ИХ)

Быстрый рост объемов информации требует особых средств для ее хранения и обработки и способствует развитию специального класса программных комплексов – информационных хранилищ (репозиторий). Единицей обработки и хранения в таких комплексах является информационный ресурс. Информационный ресурс – это файл или совокупность файлов, объединенных общим смыслом и имеющих текстовую аннотацию. В частном случае, информационный ресурс – это один или несколько текстовых файлов. Текст аннотации (или текст самого ресурса) однозначно отражает смысловое содержание данного ресурса.

Основные функции системы управления ИХ

Основными функциями информационного хранилища являются:

- хранение информационных ресурсов;
- обеспечение доступа к ресурсам с рабочих мест;
- обеспечение возможности публикации новых ресурсов;
- возможность поиска интересующих ресурсов по определенным критериям.

Самым простым примером может являться использование корпоративного файл-сервера в виде хранилища различной электронной документации. В этом случае, хранение ресурса осуществляется на файловой системе сервера. Доступ с рабочих мест организован через доступ по ftp-, http-

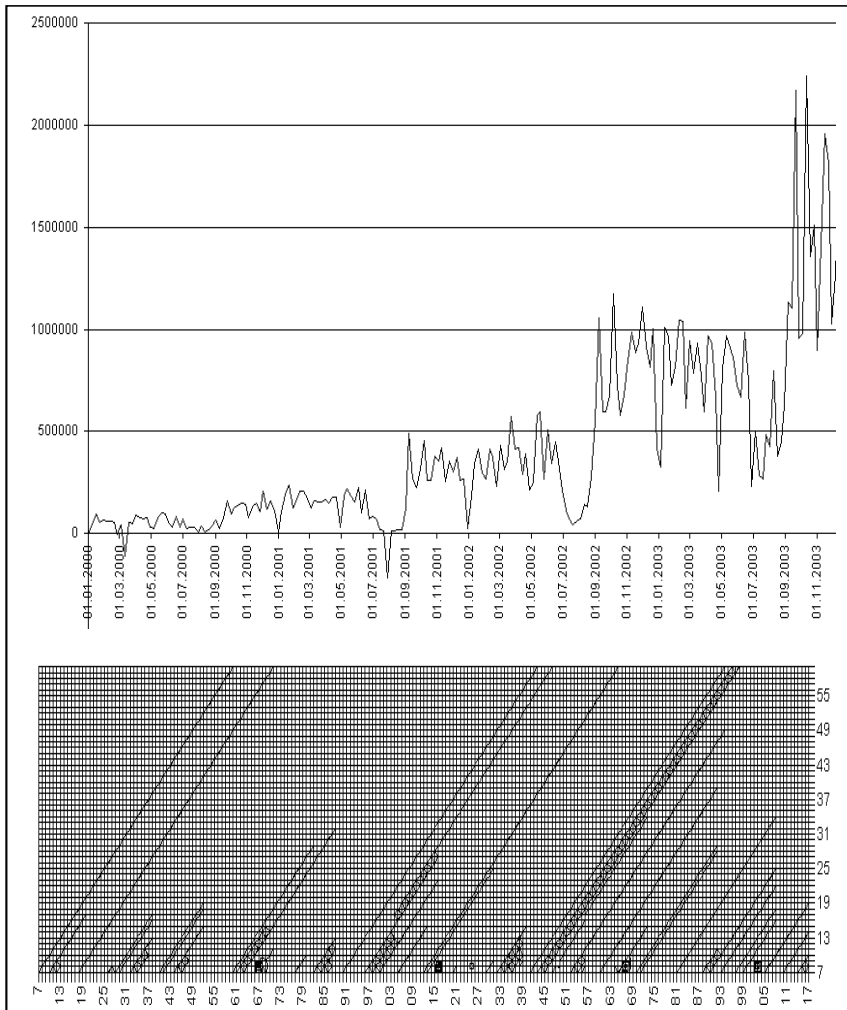


Рис. 6. Результат анализа тенденции

или smb-протоколам в локальной сети. Публикация производится путем загрузки файла/файлов на сервер по тем же протоколам. Поиск нужного ресурса происходит в виде обычного путешествия по каталогам файл-сервера.

Нейросетевая система управления ИХ

Рассмотрим проблему подробнее. По сути структура каталогов отражает структуру категорий хранимой информации. А значит, задача размещения ресурса по каталогам, есть задача классификации ресурса. При классификации ресурса должны быть решены следующие вопросы. Во-первых, должно быть составлено дерево категорий. В нашем случае, это дерево каталогов на файл-сервере, составленное группой публикаторов. Во-вторых, требуется определить, какой (каким) из этих категорий, ресурс более всего соответствует. От решения этих вопросов напрямую зависит результат классификации, а значит и последующего поиска ресурса. Но результат решения этих вопросов сильно зависит от субъективных представлений эксперта о предметной области. В случае независимой публикации ресурсов из одной предметной области разными людьми, они могут оказаться в разных категориях. Кроме того, даже у одного человека эти представления могут меняться с течением времени. Это неминуемо скажется на качестве и скорости поиска нужного ресурса конечным пользователем. Таким образом, мы не можем полагаться на субъективность человека, когда речь идет о крупном информационном хранилище с большими объемами поступающих ресурсов. Нужно какое-то унифицированное представление о соответствии ресурсов дереву категорий. Предлагаемым решением проблемы построения дерева категорий является машинная кластеризация ресурсов. При кластеризации мы полагаемся на гипотезу о том, что смысловое содержание текста можно извлечь из статистического распределения слов. То есть, по частотному распределению слов, составляющих текст ресурса (или аннотации), мы можем определить его семантику. Согласно этому предположению кластеризация основывается на данных частотного анализа текста. Процесс частотного анализа связан с морфологическим анализом слов, составляющих текст. В этом процессе все словоформы одного слова должны быть учтены как одно слово, чтобы снизить шум на входах сети. Для этого применяется механизм стемминга.

Нечеткая кластеризация информационных ресурсов с помощью сети Кохонена. Морфологический анализ текста. Стемминг

Стемминг — это формальное выделение основы слова, стабильной, графически неизменной при склонении и спряжении части слова. Задача стемминга — это задача морфологического анализа языка. Словоформа представляется в виде:

префикс + корень + суффикс + окончание + постфикс

Основу слова составляют префикс и корень. Все словоформы в процессе стемминга приводятся к их основам, и уже основы участвуют в подсчете относительных частот. По результатам подсчета частот выделяются наиболее часто встречаемые словоформы, так называемые «ключевые слова». Ключевыми словами считаются основы, которые чаще всего встретились в тексте и являются семантически значащими. Например, ключевыми словами не могут считаться предлоги, частицы, союзные слова, местоимения. Это так называемые «стоп-слова», которые отбрасываются при частотном анализе. Основываясь на результаты частотного анализа, строится нейронная сеть Кохонена, которая, собственно, и проводит кластеризацию ресурсов. На входы сети подаются относительные частоты встречаемости ключевых слов по каждому ресурсу. Сеть обучается по алгоритму «победитель получает все». Выходы сети соответствуют списку категорий. После окончания обработки всех ресурсов, сеть можно считать обученной. Для завершения построения дерева категорий, эксперту необходимо назвать образованные нейронной сетью кластеры. При этом он может основываться на наборы ключевых слов и ресурсов, отнесенных сетью к данной категории. После кластеризации задача публикации ресурсов решается этой же сетью. Она переводится в режим классификатора и выбирается наиболее подходящая для задачи функция активации. Поступающие ресурсы проходят стемминг и частотный анализ. Результаты частотного анализа подаются на входы сети. По состоянию выходов сети можно принимать решение о принадлежности ресурса к категориям. Таким образом мы избавляем публикаторов и экспертов от работы по созданию дерева категорий и принятия решений о публикации ресурса. Вопрос «куда публиковать ресурс» решен.

Классификация информационного ресурса на основе ННС

Рост объемов проектной информации хранимой в электронном виде требует развития средств поиска нужных ресурсов. Обычно используются следующие виды поиска:

- обычный текстовый поиск по подстроке с некоторыми шаблонами;
- поиск, учитывающий морфологию языка;
- нечеткий поиск;
- браунинг по категориям-каталогам.

При больших объемах и семантической разнородности хранимой информации поиска по подстроке с шаблонами и морфологического поиска уже становится недостаточно. Четкий поиск с заданными критериями сменяется так называемым «нечетким поиском», позволяющим указывать неточные значения критериев поиска. Но зачастую нет возможности точно сформулировать такие критерии. Известно только, что ресурс может принадлежать определенным (достаточно общим) категориям. В таком случае поиск по категориям в виде простого выбора категории из дерева с просмотром всех ресурсов, наполняющих ее, бывает значительно удобнее. Этот вид поиска хоть и является исторически самым старым, но до сих пор не имеет отработанных стандартов на автоматизированные средства для построения индекса. Автоматизированная индексация информационных ресурсов для последующего поиска в виде браунинга по категориям требует создания особого вида индекса. Индекс должен отражать структуру категорий и их взаимосвязь с информационными ресурсами. Для этого предложен следующий формат: $IR = (FR, NN)$, где IR — индекс ресурса; FR — относительные частоты слов текста, составляющего ресурс; NN — обученная нейронная сеть; Здесь структура категорий и их взаимосвязь с информационными ресурсами будет неявно заключена в матрице весов нейронной сети. Относительные частоты (FR) вычисляются по результатам стемминга и частотного анализа текста, составляющего ресурс. Нейронная сеть в индексе (NN) — это нейронная сеть Кохонена, обученная в процессе кластеризации ресурсов. То есть, процесс кластеризации ресурсов также дает нам и уникальный по возможностям индекс. На основе этого индекса мы можем осуществлять все вышеописанные виды поиска, включая браунинг, а также комбинировать их. Например, можно дать возможность осуществлять полнотекстовый нечеткий поиск только в определенных категориях. В тени осталась задача управления хранилищем. При последующем росте

объемов информации, поступающей в хранилище, возможно возникновение ситуации, когда текущее дерево категорий не будет более покрывать всех поступающих ресурсов. Симптомами такой ситуации будет являться появление ресурсов, не принадлежащих ни одной категории, или наоборот, принадлежащих сразу большому количеству категорий. В этом случае требуется перестройка дерева категорий с учетом вновь поступивших ресурсов, то есть повторная кластеризация всего объема ресурсов с последующим повторным именованием полученных кластеров. Итак, что нам дало наделение информационного хранилища искусственным интеллектом в виде нейронной сети?

- Информационный репозиторий сам может построить дерево категорий для хранящихся в нем ресурсов и разнести ресурсы по дереву. Теперь время экспертов требуется только для того, чтобы поименовать полученные кластеры на основе списков ключевых слов и ресурсов, отнесенных к данной категории.
- Репозиторий сам решает, к какой категории отнести тот или иной ресурс. Это позволяет публикаторам не задумываться над тем, в каких категориях опубликовать ресурс. А пользователи не становятся заложниками субъективности экспертов и публикаторов.
- Автоматизация кластеризации порождает индекс, позволяющий реализовать различные виды поиска, включая браузеринг по категориям. Это ускоряет поиск со стороны пользователей хранилища.
- Хранилище само следит за актуальностью дерева категорий. При несоответствии дерева категорий поступающим ресурсам, то есть при невозможности правильно классифицировать ресурс, хранилище должно принять решение о перекластеризации.

Таким образом от понятия «информационное хранилище» мы переходим к понятию «интеллектуальное информационное хранилище». Набор его функций стал более широким и теперь включает в дополнение к предыдущим функциям:

- импорт уже имеющихся ресурсов;
- кластеризация имеющихся ресурсов с целью составления дерева категорий;
- обработка поступающих ресурсов, то есть выполнение классификации поступающих ресурсов и публикация их в соответствующих категориях;

- возможность полнотекстового и нечеткого поиска по текстам ресурсов, возможность поиска по дереву категорий;
- слежение за тем, чтобы дерево категорий всегда покрывало поступающие ресурсы (чтобы не было ресурсов, не относящихся ни к одной категории, или наоборот, относящихся ко всем категориям).

Заключение

Перспективой развития гибридных систем считают разработку синергетического компьютера, построенного как взаимодействующая система. Вместе с тем, ННС успешно решают важные задачи в информационных системах. Для успеха применения важно разработать конкретные методики формирования обучающих выборок для ННС, выбрать архитектуры ННС, получить приемлемые настройки параметров.

Литература

1. *Хакен Г., Хакен-Крель М.* Тайны восприятия. – М.: Институт компьютерных исследований, 2002. – 272 с.
2. *Ярушклина Н. Г.* Основы теории нечетких и гибридных систем. – М.: Финансы и статистика, 2004. – 320 с.
3. *Редько В. Г.* Эволюционная кибернетика. – М.: Наука, 2001. – 156 с.
4. *Гаврилова Т. А., Хорошевский В. Ф.* Базы знаний интеллектуальных систем. – СПб: Питер, 2000. – 384 с.
5. *Ярушклина Н. Г.* Нечеткие нейронные сети с генетической настройкой // Лекция научной школы «Нейроинформатика–2004». – М.: МИФИ, 2004.
6. *Ярушклина Н. Г.* Нечеткие и гибридные системы: Обзор итогов и тенденций развития // Новости искусственного интеллекта, № 5, 2003, с. 5–12.
7. *Наместников А. М., Ярушклина Н. Г.* Эффективность генетических алгоритмов для задач автоматизированного проектирования // Известия РАН. Теория и системы управления. – № 2, 2002.
8. *Ярушклина Н. Г.* Гибридные системы, основанные на мягких вычислениях: Определение, архитектура, возможности // Программные продукты и системы. – № 3, 2002.

Н. Г. ЯРУШКИНА

Надежда Глебовна ЯРУШКИНА, доктор технических наук, заведующая кафедрой «Информационные системы» Ульяновского государственного технического университета. Область научных интересов — теория нечетких множеств и систем, искусственные нейронные сети, генетические алгоритмы, гибридные системы. Автор 2 монографий и более 160 научных публикаций.

НАУЧНАЯ СЕССИЯ МИФИ–2005

НЕЙРОИНФОРМАТИКА–2005

VII ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ

Оригинал-макет подготовлен Ю. В. Тюменцевым
с использованием издательского пакета L^AT_EX 2_ε
и набора PostScript–шрифтов PSCyr

Подписано в печать 25.11.2004 г. Формат 60 × 84 1/16
Печ. л. 13, 375. Тираж 200 экз. Заказ №

*Московский инженерно-физический институт
(государственный университет)*

Типография МИФИ

115409, Москва, Каширское шоссе, 31