МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ПО АТОМНОЙ ЭНЕРГИИ

МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ, НАУКИ И ТЕХНОЛОГИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

НАУЧНАЯ СЕССИЯ МИФИ–2004

НЕЙРОИНФОРМАТИКА–2004

**VI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ

Часть 1

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2004

УДК 004.032.26 (06)
ББК 32.818я5
М82

НАУЧНАЯ СЕССИЯ МИФИ–2004. VI ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2004»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. Часть 1. – М.: МИФИ, 2004. – 199 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 28–30 января 2004 года в МИФИ в рамках VI Всероссийской научно-технической конференции «Нейроинформатика–2004».

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 5–7262–0526–X

Содержание

Н. Г. ЯРУШКИНА
Ульяновский государственный технический университет (УлГТУ),
г. Ульяновск
E-mail: jng@ulstu.ru

НЕЧЕТКИЕ НЕЙРОННЫЕ СЕТИ С ГЕНЕТИЧЕСКОЙ НАСТРОЙКОЙ

Аннотация

Нечеткие нейронные сети представляют собой реализацию систем нечеткого логического вывода методами нейронных сетей. Такие сети включают в себя слои, состоящие из специальных *И* и *ИЛИ* нейронов. Для настройки нечетких нейронных сетей используются методы генетической оптимизации. В лекции изложено краткое введение в построение и обучение нечетких нейронных сетей. Важным преимуществом таких сетей является сочетание методов машинного обучения и вербализации правил вывода.

N. G. YARUSHKINA
Ulyanovsk State Technical University (UlSTU),
Ulyanovsk
E-mail: jng@ulstu.ru

FUZZY NEURON NETWORKS WITN GENETIC OPTIMIZATION

Abstract

Fuzzy neuron networks with genetic optimization are fuzzy logical inference systems. These nets include special *AND*, *OR* neurons. Genetic algorithms optimize a structure and parameters of fuzzy neuron networks. This survey lecture represens brief introduction to creation and learning of fuzzy neuron networks. Important practical advantage of fuzzy neuron networks is hybridization of machine learning methods and fuzzy logical rules representation in linquistic terms.

Введение

Исследования восприятия на искусственных моделях нейронных сетей успешно развивались, начиная с 40-х годов прошлого века (*МакКаллок* и *Питтс*, 1947; *Хебб*, 1949). Со знаменитой ныне статьи *Л. Заде* [20] началось постепенное развитие нечетких систем. В 1979 году *Заде* ввел теорию приближенных рассуждений [5]. Однако увлечение символьной обработкой в течение 70–80–х годов приводило к тому, что рождение и становление вычислительного интеллекта оставалось почти не замеченным. Тем не менее в 1982 году *Хопфилд* обеспечил математические основы понимания динамики нейронных сетей с обратными связями. В 1984 году *Кохонен* предложил алгоритм обучения без учителя для самоорганизующихся сетей. В 1986 году *Румельхарт* и *МакКлелленд* ввели алгоритм обратного распространения ошибки для нелинейных многослойных сетей. Книга *Холланда* «Адаптация в природе и искусственных системах» (1975) [18] заложила основы эволюционного моделирования и генетические алгоритмы стали методом структурного синтеза в интеллектуальных системах. Успехи нейронных сетей, нечетких систем и генетических алгоритмов привели к формированию направления, которое можно обозначить, как *«вычислительный интеллект»*. В результате в 1994 году в Беркли в Калифорнии *Л. Заде* ввел «зонтичный» термин *«мягкие вычисления»*, который можно пояснить следующей формулой [21]:

$$\text{Мягкие вычисления} = \textit{нечеткие системы} + \\ + \textit{нейронные сети} + \textit{генетические алгоритмы}.$$

Появление термина «мягкие вычисления» обычно объясняют тем, что мягкие или гибридные системы, такие как нечеткие нейронные сети с генетической настройкой параметров, демонстрируют взаимное усиление достоинств и погашение недостатков отдельных методов. Очевидно, что представление знаний в нейронных сетях в виде матриц весов не позволяет строить объяснение проделанного распознавания или прогнозирования, в то время как системы вывода на базе нечетких правил позволяют строить объяснения как обратные протоколы вывода (ответы на вопрос ПОЧЕМУ?). Нейронные сети обучаются с помощью универсального алгоритма, то есть трудоемкое извлечение знаний заменяется сбором достаточной по объему обучающей выборки. Для нечетких систем вывода построение включает в себя трудоемкие процессы формализации понятий, построения функций принадлежности, формирования правил вывода. А *нечеткие нейронные сети* обучаются как нейронные сети, но строят объяснения как системы нечеткого вывода. В гибридных системах генетические алгоритмы (ГА) способны выполнить настройку функций принадлежности. Функция принадлежности задается параметризованной функцией формы, параметры которой оптимизирует ГА. ГА может оптимизировать и состав больших баз нечетких продукций или структуру нейронной сети, в результате возникают *генетические нечеткие нейронные сети*. Однако названные причины формирования гибридных систем, которые составляют основное содержание вычислительного интеллекта, являются технологическими прикладными причинами.

Существует, на наш взгляд, более важная и общая фундаментальная причина развития вычислительного интеллекта. Эту причину можно обозначить как необходимость объединить в единую систему восприятие и логическую обработку. Исторически исследование восприятия (нейронные сети, распознавание образов) развивается отдельно от классического искусственного интеллекта, моделирующего логическое мышление.

Существуют разрыв между бионическим интеллектом искусственных нейронных сетей, моделирующих восприятие, и логическим интеллектом систем логического вывода. Между тем естественный интеллект, видимо, не имеет такой резкой границы. Грубую схему «слоев естественного интеллекта» можно описать следующей формулой [1]:

$$\begin{gathered}\textit{сенсорика} + \textit{моторика} + \textit{безусловные и условные рефлексы} + \\ + \textit{врожденные программы (инстинкты)} + \textit{мышление} = \text{интеллект}.\end{gathered}$$

Возможно, эффективность человеческого интеллекта состоит в том, что все такие слои работают согласованно, помогая ему решать многие недоступные искусственным системам задачи. Вычислительный интеллект объединяет в гибридные системы искусственные модели таких слоев. Сказанное позволяет рассматривать вычислительный интеллект не как технологическое достижение, а как парадигму развития искусственного интеллекта в XXI веке. Поэтому вычислительный интеллект, сочетающий восприятие (perception based theory) и мышление (computing with words), во всяком случае, уместен в решении интеллектуальных задач.

Вычислительный интеллект в широком смысле ищет и предлагает методологическую схему, содержащую неполную, нечеткую и неточную информацию, но решающую интеллектуальные задачи. Вычислительный интеллект в узком смысле ищет схемы гибридизации интеллектуальных технологий, обладающие преимуществами перед их автономным использованием. Общая задача объединения моделей восприятия и логической обработки на уровне структуры должна проявляться в том, что появляются схемы, работающие с вещественными числами (перцепция) и дискретными сигналами истины (логика). Как мы увидим ниже, схемы гибридных систем включают в себя не только классические нейроны, но и *И*, *ИЛИ*– нейроны.

Наряду с термином «мягкие вычисления» используется и другой интегрирующий термин — вычислительный интеллект. В рамках вычислительного интеллекта очень многие исследователи разрабатывают системы, объединяющие составные части мягких вычислений. В настоящее время существуют в основном нейро-нечеткие системы. Однако растет число нечетко-генетических, нейро-генетических и нейро-нечетко-генетических систем. В данной лекции анализируется эффективное использование и полезные комбинации составных частей мягких вычислений (МВ): нечеткой логики (НЛ) для контроля параметров генетических алгоритмов (ГА) и нейронных сетей (НС), ГА для формирования НС (топологии и весов), НС для настройки нечетких контроллеров.

Нечеткие системы

Нечеткие множества

Нечеткие системы опираются на лингвистическое описание отношений между переменными процессов. В отличие от базовых значений нечеткие значения представляют собой слова повседневного языка, то есть они могут описывать человеческий опыт и знания о процессах.

Пример 1. Определение множеств нечетких значений.

Объектная переменная — дата.

Множество нечетких значений:

$$F_1 = \{\text{начало недели, середина недели, конец недели}\}.$$

Функции принадлежности: Определение и смысл

В любой ситуации признак объекта проблемной области имеет одно и только одно четкое значение из согласованного множества базовых и одно

Таблица 1. Функции принадлежности для понятия «дата»

b	f_1	f_2	f_3	$\mu(b,f)$
Пн.	1.0	0.0	0.0	1.0
Вт.	0.6	0.6	0.0	1.2
Ср.	0.0	1.0	0.0	1.0
Чт.	0.0	0.6	0.3	0.9
Пт.	0.0	0.0	0.7	0.7
Сб.	0.0	0.0	0.9	0.9
Вс.	0.0	0.0	1.0	1.0

или более чем одно нечеткое значение из соответствующего множества нечетких значений. Выберем для объектной переменной «дата» множество базовых значений:

$$B = \{\text{Пн., Вт., Ср., Чт.,Пт., Сб., Вск.}\}$$

и множество нечетких значений:

$$F = \{\text{начало недели, середина недели, конец недели}\}.$$

Сущности «дата» и четкому значению «среда», соответствует нечеткое — середина недели. «Среду» с трудом можно отнести на «начало недели» или «конец недели».

Интуитивное отношение между базовым и нечетким значением объектной переменной выражают количественно с помощью *функции принадлежности*, обозначаемой греческой буквой μ.

Функция $\mu(b,f)$ отображает базовое значение b и нечеткое значение f в интервал $[0;1]$. По определению, $0 \leqslant \mu(b,f) \leqslant 1$ для b и f. Если $\mu(b_0,f_0) = 0$ для пары значений (b_0,f_0), то четкое значение b_0 не принадлежит нечеткому f_0 и не является членом f_0. Если $\mu(b_0,f_0) = 1$, то $b_0 \in f_0$. Частичная или неопределенная принадлежность выражается значением функции, лежащими в интервале между 0 и 1.

Рассмотрим таблицу значений $\mu(b,f)$ для объектной переменной «дата» (табл. 1). Сумма чисел по горизонтали часто, но не обязательно близка к 1.0. Реально это означает, что нечеткие значения не покрывают формально спектр базовых значений.

Нечеткие числа

Часто признаки объектов проблемной области с одной стороны описываются четкими числами как базовыми значениями, а с другой — нечеткими, такими как ~ 1, ~ 2.5, «грубо 6.0». Соответствующие функции принадлежности строятся как кривые Гаусса или треугольные функции.

Нечеткие интервалы

Рассмотрим объектную переменную с базовыми упорядоченными действительными числами и нечеткими значениями, заданными в виде нечетких интервалов, например: грубо между 1 и 5, приблизительно в интервале от 100 до 120, вероятно, между 5 и 6. Функция принадлежности обычно строится как трапециевидная по форме.

Операции с нечеткими множествами

Дополнительное множество *НЕ*. Рассмотрим определение операции дополнения в классической теории множеств. Пусть M — нечеткое множество элементов из множества базовых значений B, тогда дополнительное множество $\neg M$, определяется как множество из тех базовых элементов B, которые не принадлежат множеству

$$M : \neg M = \{b;\ \ b \in B, b \notin M\}.$$

Определим дополнение в теории нечетких множеств. Рассмотрим нечеткое множество $M = \{(b, \mu);\ \ b \in B, 0 \leqslant \mu \leqslant 1\}$, дополнительное нечеткому множеству $\neg M$, определяемому как

$$\neg M = \{(b, (1 - \mu)),\ \ b \in B, 0 \leqslant \mu \leqslant 1\}.$$

Пересечение *И*. Рассмотрим определение операции пересечения в традиционной теории множеств. Пусть заданы два множества M_1 и M_2 элементов из множества базовых значений B, тогда пересечение множеств $M = M_1 \cap M_2$ определяет множество, элементы которого из B принадлежат к обоим множествам M_1 и M_2 одновременно

$$M = M_1 \cap M_2 = \{b;\ \ b \in M_1 \text{ и } b \in M_2\}.$$

Определим операцию пересечения в теории нечетких множеств. Пусть заданы два нечетких множества

$$M_1 = \{(b, \mu_1) \ |b \in B, 0 \leqslant \mu_1 \leqslant 1\},$$
$$M_2 = \{(b, \mu_2) \ |b \in B, 0 \leqslant \mu_2 \leqslant 1\}.$$

Пересечение нечетких множеств определяется как

$$M = M_1 \cap M_2 = \{(b, \ \min(\mu_1, \mu_2))\}.$$

Значение $b_0 \in M_1$ и $b_0 \in M_2$ полностью содержится в $M_1 \cap M_2$:

$$\mu_1(b_0) = 1; \quad \mu_2(b_0) = 1 \Rightarrow \min(\mu_1, \mu_2) = 1.$$

Значение b_0, частично содержащееся в M_1 или в M_2, только частично содержится в $M_1 \cap M_2$:

$$0 \leqslant \mu_1(b_0) \leqslant 1; \ 0 \leqslant \mu_2(b_0) < 1 \Rightarrow 0 \leqslant \min(\mu_1, \mu_2) \leqslant 1.$$

Если значение $b_0 \notin M_1$ и $b_0 \notin M_2$, то оно не содержится в

$$M_1 \cap M_2 : \mu_1(b_0) = 0, \ \mu_2(b_0) = 0 \Rightarrow \min(\mu_1, \mu_2) = 0.$$

Объединение *ИЛИ* . Рассмотрим определение операции объединения в традиционной теории множеств. Пусть даны два множества M_1 и M_2 элементов базового множества B, тогда объединением множеств $M = M_1 \cup M_2$ называется множество, содержащее такие элементы B, которые принадлежат множествам M_1 или M_2 или обоим одновременно:

$$M = M_1 \cup M_2 \Rightarrow \{b; \ b \in M_1 \text{ или } b \in M_2\}.$$

Определим объединение в теории нечетких множеств. Пусть даны два нечетких множества

$$M_1 = \{(b, \mu_1)| \ b \in B, \ 0 \leqslant \mu_1 \leqslant 1\},$$
$$M_2 = \{(b, \mu_2)| \ b \in B, \ 0 \leqslant \mu_2 \leqslant 1\},$$

тогда объединением множеств $M = M_1 \cup M_2$ называется

$$M = \{(b, \max(\mu_1, \mu_2))| \ b \in B, \ 0 \leqslant \mu_1 \leqslant 1, \ 0 \leqslant \mu_2 \leqslant 1\}.$$

Для заданного множества $F = \{f\}$ нечетких значений объектной переменной A, мы можем добавить новые термины, используя связку *ИЛИ*. Функция принадлежности нового значения объектной переменной A определяется как:

$$\mu(b, f_1 \textit{ ИЛИ } f_2) = \max[\mu(b, f_1), \mu(b, f_2)], \quad \forall f_1 \textit{ ИЛИ } f_2 \in A.$$
$$M(f_1 \textit{ ИЛИ } f_2) = \{b \mid \ \mu(b, f_1 \textit{ ИЛИ } f_2)\} =$$
$$= \{b \mid \ \max(\mu(b, f_1), \mu(b, f_2))\} = M(f_1) \cup M(f_1).$$

Следовательно, нечеткое множество, принадлежащее к нечетким значениям f_1 *ИЛИ* f_2, это нечеткая дизъюнкция множеств, принадлежащих f_1 и f_2.

Обобщенные определения пересечения и объединения нечетких множеств . Выше мы определили операции пересечения и объединения двух нечетких множеств следующим образом:

- пересечение:

$$M_1 \cap M_2 = \{b \mid \ \min(\mu_1, \mu_2)\},$$

- объединение:

$$M_1 \cup M_2 = \{b \mid \ \max(\mu_1, \mu_2)\}.$$

Фактически эти определения — специальные случаи общего, использующего понятия T–норм и S–конорм [19]. Эти выражения обозначают функции со специальными свойствами, которые будут детально обсуждаться далее. T–нормы и S–конормы обладают двумя важными свойствами. $T(x, y)$ и $S(x, y)$ — это функции двух действительных переменных x и y, определенных на интервалах $[0, 1] \times [0, 1]$. Значения функций $T(x, y)$ и $S(x, y)$ также находятся в интервале $0 \leqslant T(x, y) \leqslant 1$ и $0 \leqslant S(x, y) \leqslant 1$. Используя любые T–нормы $T(x, y)$ и S–конормы $S(x, y)$, обобщим определения операций над нечеткими множествами:

- пересечение:

$$M_1 \cap M_2 = \{b \mid \ T(\mu_1, \mu_2)\},$$

Таблица 2. T–нормы

$T(x,y)$	Имя функции
$y = \begin{cases} 0 & \text{, если} \quad \max(x,y) > 1 \\ \min(x,y) & \text{, если} \quad \max(x,y) \leqslant 1 \end{cases}$	ограниченное произведение
$\max(0, x+y-1)$	усиленная разность
$\frac{x*y}{1+(1-x)*(1-y)}$	произведение Эйнштейна
$x*y$	алгебраическое произведение
$\frac{x*y}{1-(1-x)*(1-y)}$	произведение Гамахера
$\min(x,y)$	минимум

- объединение:

$$M1 \cup M_2 = \{b \mid\ S(\mu_1, \mu_2)\}.$$

- операция *И*:

$$\mu(b, f_1 \,\textit{И}\, f_2) = T(\mu(b, f_1), \mu(b, f_2)).$$

- операция *ИЛИ*:

$$\mu(b, f_1 \,\textit{ИЛИ}\, f_2) = S(\mu(b, f_1), \mu(b, f_2)).$$

Далее будет показано что для $0 \leqslant x \leqslant 1$ и $0 \leqslant y \leqslant 1$ функция $\min(x,y)$ соответствует T–норме и функция $\max(x,y)$ — S–конорме. В табл. 2 и 3 перечислены некоторые широко известные T–нормы и S–конормы [3].

Общие свойства T-норм и S-конорм

T–нормы и S–конормы имеют следующие общие свойства:

- интервалы определения:

$$0 \leqslant x \leqslant 1, \ \ 0 \leqslant y \leqslant 1;$$

- интервал значений:

$$0 \leqslant T(x,y) \leqslant 1, \ \ 0 \leqslant S(x,y) \leqslant 1;$$

Таблица 3. S–конормы

$S(x, y)$	Имя функции
$\max(x, y)$	максимум
$1 - (1 - x) * \frac{(1-y)}{(1-x*y)}$	сумма Гамахера
$1 - (1 - x) * (1 - y)$	алгебраическая сумма
$1 - (1 - x) * \frac{(1-y)}{(1+x*y)}$	сумма Эйнштейна
$\min(1, x + y)$	ограниченная сумма
$y = \begin{cases} 1 & \text{, если} \quad \min(x, y) > 0 \\ \max(x, y) & \text{, если} \quad \min(x, y) = 0 \end{cases}$	усиленная сумма

- коммутативность:

$$T(x, y) = T(y, x), \;\; S(x, y) = S(y, x);$$

- ассоциативность:

$$T[T(x, y), z] = T[x, T(y, z)],$$
$$S[S(x, y), z] = S[x, S(y, z)];$$

- монотонность:

$$\text{Если } x = u \text{ и } y = v, \text{ то } T(x, y) = T(u, v), \;\; S(x, y) = S(u, v);$$

- специальные значения:

$$T(0, y) = 0,$$
$$S(0, y) = y,$$
$$T(1, y) = y,$$
$$S(1, y) = 1.$$

Каждая функция со свойствами T называется T–нормой или треугольной нормой, а со свойствами S — S–конормой или триангулярной конормой. Каждая пара триангулярных норм и конорм, которые подчиняются обобщенным отношениям Де-Моргана, называется парой относительных триангулярных норм. Все функции принадлежности (как для *И*, так и для *ИЛИ*–связок нечетких значений) выбираются такими же далекими друг от друга, как пары относительных триангулярных T–норм и S–конорм. T–нормы выбираются для *И*, S–конормы — для *ИЛИ*. Функции называют триангулярными потому, что соотношения

$$T(x,y) = T(y,x) \text{ и } S(x,y) = S(y,x)$$

выполняются только в треугольной области $0 \leqslant x \leqslant 1, 0 \leqslant y \leqslant x$.

Нечеткие отношения

Общие определения. В практических приложениях бывает полезно соединить два стандартных нечетких множества в одно комбинированное нечеткое множество, называемое *декартовым произведением*. Подобная концепция широко используется в традиционной теории множеств.

Предположим имеется два объекта проблемной области A и B с двумя множествами базовых значений $\{a\}$ и $\{b\}$, нечетких значений $\{f\}$ и $\{g\}$, и функциями принадлежности $p(a,f)$ и $q(b,g)$, соответственно. Сущности A и B могут быть как различными, так и фактически идентичными.

Рассмотрим два нечетких множества $P(f) = \{a|p(a,f)\}$ объектной переменной A и $Q(g) = \{b|q(b,g)\}$ переменной B. Операция $n(x,y)$ будет нормой или конормой, соответственно. Определим произведение множеств $P(f)$ и $Q(g)$ следующим образом:

$$\begin{gathered} Z(f,g) = P(f) \otimes Q(g) = \{(a,b)|\ n[p(a,f), q(b,g)]\}, \\ \{a, p(a,f)\} \in P(f),\ \ \{b, q(b,g))\} \in Q(g). \end{gathered}$$

Другими словами, Z — это множество всех триплетов a, b, n, где a и b базовые значения A и B, f и g нечеткие значения A и B, а $n[p(a,f), q(b,g)] = \mu[(a,b),(f,g)]$ определяет комбинированную принадлежность базовых значений (a,b) нечетким значениям (f,g).

Если пара (f,g) нечетких значений связывается связкой *И*, т. е. $(f,g) = f$ И g, то требуется выбрать T–норму $n(x,y)$. Если $(f,g) = f$ ИЛИ g, то необходимо связывать f и g функцией $n(x,y)$, которая должна быть S–конормой.

Двухместные нечеткие множества. Нечеткое бинарное отношение. Пусть X и Y — непустые множества. Нечеткое отношение R — это нечеткое подмножество $X \otimes Y$. Другими словами, $R \in F(X \otimes Y)$. Если $X = Y$, то говорят, что R — это бинарное нечеткое отношение на X. Если R — это бинарное нечеткое отношение на $X \otimes X$, тогда $R(x, y)$ можно интерпретировать как степени принадлежности упорядоченных пар (x, y).

Пример 2. Бинарное нечеткое отношение.

Бинарное нечеткое отношение на множестве $U = \{1, 2, 3\}$ назовем «приблизительно равно»

$$R(1,1) = R(2,2) = R(3,3) = 1,$$
$$R(1,2) = R(2,1) = R(2,3) = R(3,2) = 0.6,$$
$$R(1,3) = R(3,1) = 0.2.$$

Значимость нечетких отношений обусловлена тем, что они описывают взаимодействие между переменными.

Операции над нечеткими отношениями. Пусть R и S — два бинарных нечетких отношения на $X \times Y$. Пересечение R и S определяется следующим образом:

$$(R \cap S)(x, y) = \min(R(x, y), S(x, y)),$$

где $R, S : X \times Y \to [0, 1]$.

Объединение R и S определяется как

$$(R \cup S)(x, y) = \max(R(x, y), S(x, y)).$$

Проекции двухместных функций принадлежности. Если в двухместных функциях принадлежности $[(a, b), (f, g)] = n[p(a, f), q(b, g)]$, переменные b и g зафиксировать на уровне значений $b = b_0$ и $g = g_0$ объектной переменной B, то функция $\nu(a, f) = \mu[(a, b_0), (f, g_0)]$ называется *проекцией* $\mu[(a, b), (f, g)]$ на значения (b_0, g_0) переменной B. Функция $\nu(a, f)$ зависит только от двух переменных a, f и представляет собой принадлежность базовых значений объектной переменной A к нечетким, причем имеется столько проекций $\nu(a, f)$ двухместных функций принадлежности $\mu[(a, b), (f, g)]$, сколько возможно комбинаций (b_0, g_0). Проекцией нечеткого отношения является тоже нечеткое множество. Для того чтобы задать проекции, необходимо указать их функции принадлежности. Нечеткие проекции [15] нечеткого отношения определяются следующим образом:

$$\Pi_X R(x, y) = \sup R(x, y) \mid \ y \in Y,$$
$$\Pi_Y R(x, y) = \sup R(x, y) \mid \ x \in X.$$

Определение проекций позволит в дальнейшем определить композицию нечетких отношений и композицию нечеткого множества с нечетким отношениям.

Графически проекция нечеткого отношения приведена на рис. 1.

Нечеткое множество A нормально тогда и только тогда, когда $\sup_x \mu_A(x) = 1$. Если A и B — нормальные множества, то

$$\begin{aligned}
\Pi_y(A \otimes B) &= B, \\
\Pi_x(A \otimes B) &= A, \\
\Pi_x(x) &= \sup\big\{(A \otimes B)(x, y) \ \mid y \in Y\big\} = \\
&= \sup\big\{A(x) \cap B(y) \ \mid y \in Y\big\} = \\
&= \min\big\{A(x), \sup\{B(y)\} \ \mid y \in Y\big\} = \\
&= \min\big\{A(x), 1\big\} = A(x).
\end{aligned}$$

Функции нечетких переменных

Функции с одной независимой переменной. Для описания проблемной области используются либо четкие числа x из некоторого множества четких $\{x\}$, либо нечеткие числа f из некоторого множества нечетких значений $\{f\}$. Представления связаны друг с другом функцией принадлежности $\mu(x, f)$ соответствующего нечеткого множества $X(f) = \{(x, \mu(x, f))\}$.

Функция $w = \mathrm{fun}(x)$ может быть задана алгоритмом, который каждому четкому числу x из $\{x\}$ ставит в соответствие только одно четкое число w из определенного множества $\{w\}$ действительных чисел. Функция $w = \mathrm{fun}(x)$ — это «образ» x после выполнения преобразования $\mathrm{fun}(x) : x \to w$. Существует ли подобное определение функции для нечетких чисел? Пусть f_0 — некоторое нечеткое число (например, f_0 = «около 3.4»), а $X(f_0) = \{(x|\mu(x, f_0)\}$ — соответствующее нечеткое множество. Тогда найдем нечеткое множество W.

- Пусть (x_0, μ_0) — любой элемент нечеткого множества $X(f_0)$, тогда четкое значение w_0 определяется по формуле $w_0 = \mathrm{fun}(x_0)$.
- Принадлежность $r(w_0)$ вычисляется следующим образом:
  - соберем все $(x, \mu) \in X(f_0)$ с $\mathrm{fun}(x) = w_0$, пусть этими элементами будут (x_0, μ_0), $(x_1, \mu_1), \ldots, (x_N, \mu_N)$,
  - вычислим $r_0 = S(\mu_0, \mu_1, \ldots, \mu_N)$ по некоторой ранее согласованной S–конорме,

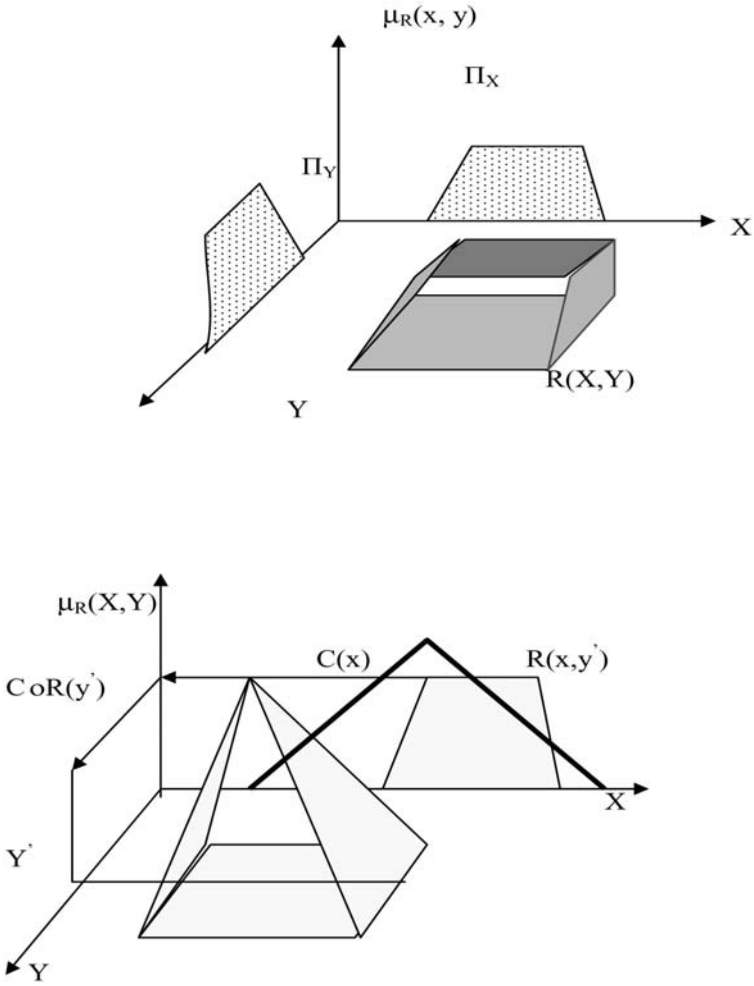


Рис. 1. Проекция и композиция нечеткого отношения

- В результате $W = \{(w,r)\}$ будет множеством всех пар $(w,r) = (w_0,r_0)$, вычисленных для всех возможных элементов $(x_0,m_0) \in X(f_0)$.

Рассмотрим W как нечеткое множество нечетких чисел p_0, определенных на множестве $\{w\}$ базовых значений w. Нечеткое число p_0 рассматривается как «образ» $p_0 = \mathrm{fun}(f_0)$. Так W получает значения $W(\mathrm{fun}(f_0))$. В общем случае для любого нечеткого числа из $\{f\}$ справедливо:

$$x \to w = \mathrm{fun}(x),\ f \to p = \mathrm{fun}(f),\ X(f) \to W(p).$$

Если нечеткое множество W получено из нечеткого множества X, то говорят, что существует отношение между X и W или $X \to W$ [13].

Композиция нечеткого множества и отношения. Выразим процедуру вычисления функции от нечеткого значения с помощью формул операций $\sup - \min$ [15]. Композиция $\sup - \min$ нечеткого множества $C \in F(X)$ и нечеткого отношения $R \in F(X \otimes Y)$ определяется как

$$(C \circ R)(y) = \sup_{x \in X} \min\{C(x), R(x,y))$$

для всех $y \in Y$.

Композицию нечеткого множества C и нечеткого отношения R можно представить как тень отношения R на нечеткое множество C (рис. 1).

Пусть A и B — нечеткие числа, а $R \subset A \otimes B$ будет нечетким отношением. Очевидны следующие свойства композиции

$$A \circ R = A \circ (A \otimes B) = A,$$
$$B \circ R = B \circ (A \otimes B) = B,$$

Определим $\sup - \min$ композицию нечетких отношений. Пусть $R \in (X \otimes Y)$ и $S \in F(Y \otimes Z)$. Тогда $\sup - \min$ композиция R и S, обозначаемая $R \circ S$, определяется как

$$(R \circ S)(u,w) = sup_{x \in X}\{R(u,v), S(v,w)\}.$$

Очевидно, что $R \circ S$ — бинарное нечеткое отношение на X и Z (рис. 1).

Нечеткие системы

Определение лингвистической переменной. Системы нечетких продукций строятся на основе понятия лингвистической переменной, которой называют пятерку объектов:

$$(x, T(x), U, G, M),$$

где x — собственное имя переменной; $T(x)$ — терминальное множество, т. е. набор значений переменной (нечетких меток); U — множество объектов (или универсум); G — синтаксические правила употребления; M — семантические правила употребления [8]. Примерами лингвистических переменных служат: количество денежных средств в финансовых решениях; температура, давление воздуха в технических процессах; средняя отметка в процессах образования; плотность населения в социальных процессах. Количество денежных средств может быть выражено как: большое, малое, 1000 рублей, значительное, незначительное; динамика давления воздуха — выросло, осталось прежним, максимально низкое, 1023 Ра; отметка — отлично, хорошо, удовлетворительно, неудовлетворительно; численность населения — значительная, средняя, малая.

Схема приближенного логического вывода. Задача интерполяции. Рассмотрим логический вывод по нечетким продукциям на основе теории приближенных вычислений. Пусть

$$\begin{aligned} y &= f(x) \ \text{— предпосылка,} \\ x &= x' \ \text{— факт,} \\ y' &= f(x') \ \text{— заключение.} \end{aligned}$$

Предпосылку рассуждения $f(x)$ зададим с помощью N значений функции:

$$\begin{aligned} R_1 &: \ \text{если} \ x_1, \ \text{то} \ y_1; \\ R_2 &: \ \text{если} \ x_2, \ \text{то} \ y_2; \\ &\qquad \dots \\ R_n &: \ \text{если} \ x_n, \ \text{то} \ y_n. \end{aligned}$$

Текущее значение $x = x'$.

Необходимо найти значение y', зависящее от x'. Нечеткая задача интерполяции формулируется с использованием зависимостей нечетких множеств:

$$\begin{aligned} R_1: &\ \text{если}\ A_1(x_1),\ \text{то}\ B_1(y_1); \\ R_2: &\ \text{если}\ A_2(x_2),\ \text{то}\ B_2(y_2); \\ &\ \dots \\ R_n: &\ \text{если}\ A_n(x_n),\ \text{то}\ B_n(y_n); \end{aligned}$$

и нечеткого значения входной переменной — $C(x')$.

Необходимо найти значение $D(y')$, входящего в заключение.

Обобщив нечеткую зависимость переменных x и y, получим продукцию: если $A(x)$, то $B(y)$. Цель нечеткого логического вывода — определить значения выходной переменной $B'(y)$ от входной переменной $C(x')$. Результирующее нечеткое множество $B' = C \circ (A \to B)$ — композиция нечеткого множества C и импликации нечетких множеств $A \to B$. Нечеткое множество B' заключения вычисляется как композиция двух нечетких множеств — факта и импликации. Композиция факта и импликации представляет собой проекцию импликации на факт.

$$A' \circ (A \to B) = \sup \min(A', A \to B);\ \ y \in Y.$$

На основании задачи нечеткой интерполяции строят оболочку экспертной системы, осуществляющей вывод на словах [15].

Основные правила умозаключений. 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):

$$\text{если}\ \ A(x) \to B(x),\ A'(x),\ \ \text{то}\ \ B'(x).$$

Нечеткий modus tollens (FMT):

$$\text{если}\ \ A(x) \to B(x),\ \neg B'(x),\ \ \text{то}\ \ \neg A'(x).$$

При использовании систем нечеткого вывода особое значение имеет *дефазификация,* т. е. определение четкого представляющего элемента по нечеткому множеству. Самым распространенным методом дефазификации является определение центра тяжести фигуры, образуемой функцией принадлежности и осью абсцисс (центроидный метод). С его помощью вычисляют центр тяжести фигуры:

$$\frac{\int_w z * C(z)dz}{\int_W C(z)dz} \Rightarrow \frac{\sum_{i=1}^n z_i * C(z_i)}{\sum_{i=1}^n C(z_i)}.$$

На практике используют метод выбора максимума:

$$Z = \min\{Z|C(Z) = \max(C(Z))\}.$$

В некоторых системах применяют также методы минимального и среднего максимума. Правильно выбранная дефазификация существенно влияет на эффективность работы системы нечеткого вывода.

Универсальная аппроксимация с помощью систем нечеткого вывода. Свойство универсальности применения систем нечеткого вывода доказано следующей фундаментальной теоремой. В 1992 году *У. Ванг* показал, что справедливо следующее утверждение: если нечеткая импликация основана на использовании операции минимум, функция принадлежности задается гауссовым распределением:

$$\mu(t) = \exp(-1/2 * ((a - t^2)/b))$$

и используется центроидный метод дефазификации, то система нечеткого вывода является универсальным аппроксиматором.

В 1995 году *К. Кастро* доказал справедливость следующей теоремы: если импликация основана на использовании операции произведения по Ларсену, а функции принадлежности — треугольные, то при использовании центроидной дефазификации нечеткий контроллер является универсальным аппроксиматором [11].

Схемы нечеткого вывода Рассмотрим различные схемы вывода на базе нечетких правил. Пусть входная переменная x является каким-либо нечетким множеством A, и y — нечетким множеством B, тогда выходная переменная z является нечетким множеством C. Если задан факт $x = x_0, y = y_0$, то необходимо найти $z = z_0$.

Схема вывода на базе нечетких правил сводится к решению следующей задачи:

База правил

$$
\begin{aligned}
&R_1: \ \text{если} \ (X \text{ is } A_1) \ \textit{И} \ (Y \text{ is } B_1) \ \text{тогда} \ (Z \text{ is } C_1),\\
&\quad \text{в противном случае}\\
&R_2: \ \text{если} \ (X \text{ is } A_2) \ \textit{И} \ (Y \text{ is } B_2) \ \text{тогда} \ (Z \text{ is } C_2),\\
&\quad \text{в противном случае}\\
&\qquad\qquad \ldots\\
&R_n: \ \text{если} \ (X \text{ is } A_n) \ \textit{И} \ (Y \text{ is } B_n) \ \text{тогда} \ (Z \text{ is } C_n).
\end{aligned}
$$

Факт: $x = x_0, y = y_0$.

Следствие $z = ?$

Для реализации интеллектуальной системы логического вывода по базе нечетких правил, необходимо определить функции:

- представления в системе нечетких понятий (функций принадлежности);
- вычисления логических выражений условных частей правил с логическими связками *И*, *ИЛИ*;
- вычисления импликации;
- усреднения результата, получаемого по разным правилам путем композиции.

Каждая из четырех перечисленных выше функций задает вариативность схемы нечеткого вывода.

Интеллектуальная система осуществляет логический вывод по базе нечетких правил по этапам.

- Фазификация 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.

  $$((x_0 \circ A_i) \cap (y_0 \circ B_i)) \to C_i$$

  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.

$$C = \cup_{i=1}^{n} C_i.$$

- Дефазификация, т. е. выбор представляющего элемента по агрегированному нечеткому понятию.

Схемы нечеткого вывода у разных авторов уточняются тем не менее только до оператора нечеткой импликации. Распространены пять схем нечеткого вывода [15]. В *схеме 1 нечеткого вывода Мамдани* импликация моделируется минимумом, а агрегация — максимумом (рис. 2).

Схема 2 разработана *Цукамото* для монотонных функций принадлежности.

На рис. 2 использованы следующие обозначения: L'_1, L'_2 — уровни достоверности применения правил, z_1, z_2 — значения выходной переменной по первому и второму правилу,

$$\begin{aligned} z'_1 &= C_1^{-1}(L_1), \\ z'_2 &= C_2^{-1}(L_2), \\ z &= \frac{L'_1 z_1 + L'_2 z_2}{L'_1 + L'_2}. \end{aligned}$$

Ввиду монотонности функций вычисления выходной переменной сводят к усреднению значений, полученных по разным правилам.

Схема 3 по *Суджено* ограничивает правые части правил вывода линейным случаем:

$$\begin{aligned} &\text{Если } (x \text{ is } A_1) \ \textit{И} \ (y \text{ is } B_1) \text{ тогда } (z = a_1 * x + a_2 * y), \\ &\text{Если } (x \text{ is } A_2) \ \textit{И} \ (y \text{ is } B_2) \text{ тогда } (z = b_1 * x + b_2 * y). \end{aligned}$$

Схема 4 по *Ларсену* выполняет импликацию с помощью произведения.

Схема 5 — это упрощенная схема нечеткого вывода:

$$\text{Если } (x \text{ is } A_i) \ \textit{И} \ (y \text{ is } B_i) \text{ тогда } (z = Z_i),$$

где Z_i — четкое значение.

Так как правые части правил в упрощенной схеме логического вывода задаются четко, то в результате вывода получается дискретное множество решений, для каждого элемента которого задана определенная степень уверенности. В качестве значений выходной переменной выбирается значение с максимальной уверенностью.

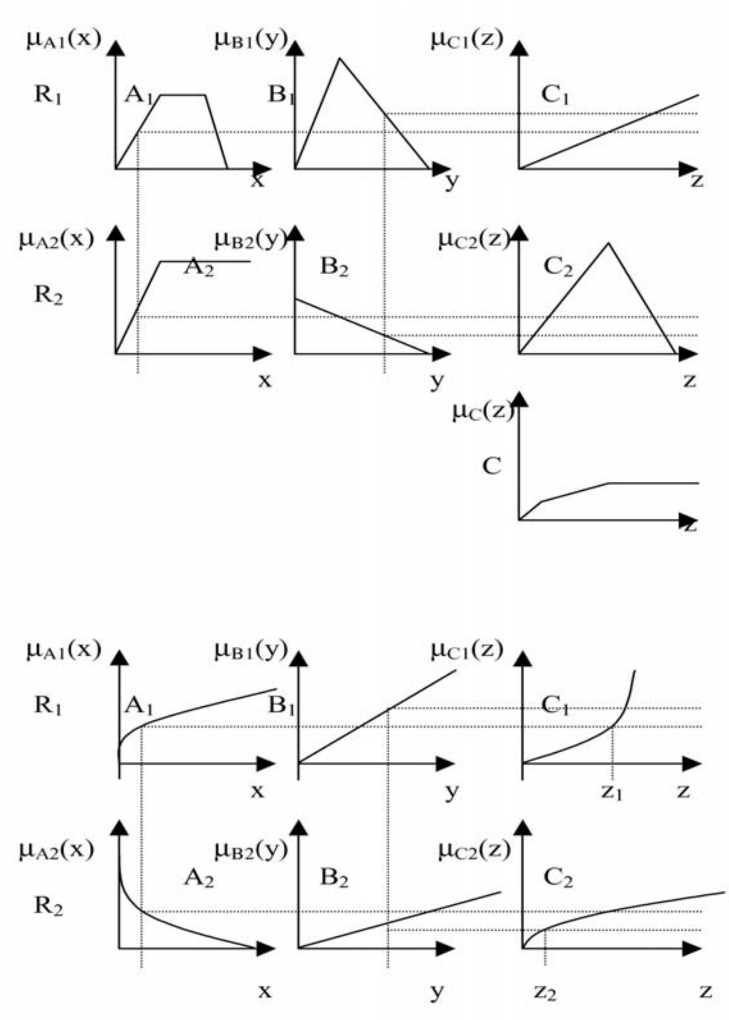


Рис. 2. Схемы нечеткого вывода по Мамдани и Цукамото

Генетические вычисления

Генетические алгоритмы

Генетические алгоритмы — это адаптивные методы поиска, которые в последнее время часто используются для решения задач функциональной и структурной оптимизации. Генетическими они называются потому, что строятся на принципах эволюции биологических организмов *Ч. Дарвина*. Популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора, т. е. принципу *«выживает наиболее приспособленный»* (survival of the fittest). Генетические алгоритмы (ГА) способны «развивать» решения реальных задач, если те закодированы соответствующим образом. ГА могут использоваться для проектирования структуры механизмов, поиска оптимальной формы детали, раскроя ткани. Например, израильская компания Schema на основе ГА разработала программный продукт Channeling для оптимизации работы сотовой связи путем выбора оптимальной частоты, на которой будет вестись разговор.

Основные принципы ГА были сформулированы *Дж. Голландом*. ГА приближенно моделирует процессы, происходящие в популяциях. В природе особи в популяции конкурируют друг с другом за различные ресурсы, за возможность породить потомство. Наиболее приспособленные к окружающим условиям особи будут иметь больше шансов воспроизвести потомков. Следовательно, гены высоко адаптированных особей распространяются в популяции и количество их потомков возрастает в каждом последующем поколении, т. е. вид развивается за счет приспособления к среде обитания. ГА по аналогии с эволюционным механизмом работают с популяцией, каждая из хромосом которой представляет возможное решение данной задачи. Каждая хромосома оценивается мерой ее *«приспособленности»* (fitness function), которую называют также *функцией оптимальности*. В частности, мерой приспособленности хромосом, кодирующих монтажную схему радиотехнического изделия, может служить длина проводников в задаче размещения элементов. Наиболее приспособленные особи получают возможность «воспроизвести» потомство с помощью *«перекрестного скрещивания»* (recombination) с другими особями популяции. В результате появляются новые особи, сочетающие в себе характеристики, наследуемые ими от родителей. Наименее приспособленные особи постепенно исчезают из популяции в процессе эволюции. Новое поколение обладает лучшими характеристиками по сравнению с предыдущим. Скрещивание наиболее приспособленных особей приводит к тому, что эволюция отыскивает перспективные решения в широком пространстве поиска. В конечном итоге, популяция сходится к оптимальному решению задачи. Существует много способов реализации идеи биологической эволюции в рамках ГА. Традиционным считается ГА, представленный ниже с использованием псевдокода. Большими буквами записаны управляющие операторы алгоритма, а курсивом — генетические операторы [4, 10]:

НАЧАЛО — генетический алгоритм
    Создать начальную популяцию
    Оценить приспособленность каждой особи
    останов := ЛОЖЬ
ПОКА НЕ останов **ВЫПОЛНЯТЬ**
НАЧАЛО — создать популяцию нового поколения
ПОВТОРИТЬ (размер популяции / 2) **РАЗ**
НАЧАЛО — цикл воспроизводства
    Выбрать две особи с высокой приспособленностью
        из предыдущего поколения для скрещивания
    Скрестить выбранные особи и получить двух потомков
    Оценить приспособленности потомков
    Поместить потомков в новое поколение
КОНЕЦ
ЕСЛИ приспособленность лучшего потомка > порога **ТО**
    останов := ИСТИНА
КОНЕЦ
КОНЕЦ

Применение генетических алгоритмов

ГА применяются для решения многих конкретных научных и технических проблем, но самое популярное приложение генетических алгоритмов — оптимизация многопараметрических функций. Эффективность ГА заключена в его способности манипулировать одновременно многими параметрами, которая использовалось во многих прикладных программах, включая проектирование интегральных схем, настройку параметров алгоритмов, поиск решений нелинейных дифференциальных уравнений и др.

Эффективность применения ГА [9] зависит от характеристик задачи. Если пространство поиска в задаче небольшое, то решение можно найти методом полного перебора и ГА не нужен для решения задачи. Если пространство гладкое и унимодальное, то любой градиентный алгоритм будет эффективнее ГА. Если существует дополнительная информация о пространстве поиска, то методы поиска, использующие эвристики, часто превосходят ГА.

Если существует дополнительная информация о пространстве поиска, то методы поиска, использующие эвристики, часто превосходят ГА. Область применения ГА — задачи с большим пространством поиска решений и отсутствием эвристической информации. Эффективность ГА зависит не только от параметров задачи, но и от характеристик метода кодировки решений, выбора генетических операторов, настройки параметров.

Стандартный ГА

Для реализации ГА сначала выбирают подходящую структуру для представления оптимизируемых решений. С точки зрения задачи поиска, одно решение представляет точку в пространстве поиска всех возможных решений. Популяция решений состоит из одной или большего количества хромосом. Обычно хромосома — это битовая строка, хотя ГА используют не только бинарное представление. Хромосомы строят с использованием векторов вещественных чисел, строк фиксированной или переменной длины. Каждая хромосома представляет собой конкатенацию ряда подстрок, называемых генами. Гены располагаются в различных позициях или локусах хромосомы, и принимают значения, называемые аллелями, т. е. ген — бит, локус — его позиция в строке, и аллель — его значение (0 или 1). Биологический термин «генотип» относится к полной генетической модели и соответствует структуре хромосомы в ГА. Термин «фенотип» относится к внешним наблюдаемым признакам и соответствует вектору в пространстве параметров. Обычно, методика кодирования реальной переменной x состоит в ее преобразовании в двоичные целочисленные строки заданной длины.

С помощью пропорционального отбора назначают каждой i-ой хромосоме вероятность $P_s(i)$, равную отношению ее приспособленности к суммарной приспособленности популяции:

$$P_s(i) = \frac{f(i)}{\sum_{i=1}^{n} f(i)}$$

Отбор и замещение всех n особей происходит в соответствии с величиной $P_s(i)$. Простейший оператор пропорционального отбора — *«рулетка»* (roulette-wheel selection), которая отбирает особей с помощью n «запусков» рулетки. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер i-ого сектора пропорционален соответствующей величине $P_s(i)$. При таком отборе члены популяции с более высокой приспособленностью выбираются чаще, чем особи с низкой. ГА выполняет оператор рекомбинации (кроссовер) для n выбранных особей с заданной вероятностью P_c, для чего n строк разбиваются на $(n/2)$ пар случайным образом.

Для каждой пары применяется кроссовер с вероятностью P_c. Выражение $(1 - P_c)$ выражает вероятность сохранения особей, которые переходят на стадию мутации. Потомки, полученные с помощью кроссовера, заменяют собой родителей и также переходят к мутации. Одноточечный кроссовер работает следующим образом. Сначала, случайным образом выбирается одна из $(l - 1)$ точек разрыва. Точка разрыва — граница между соседними битами в строке. Родительские структуры разрываются в этой точке на два сегмента. Затем, соответствующие сегменты различных родителей склеиваются и получаются два генотипа потомков.

После стадии кроссовера выполняются операторы мутации, которые с вероятностью P_m изменяют случайный бит в каждой строке на противоположный. Популяция, полученная после мутации, замещает старую и обработка одного поколения завершается. Последующие поколения обрабатываются в вышеописанной последовательности: отбор, кроссовер и мутация. В настоящее время используют разные операторы отбора, кроссовера и мутации. Например, турнирный отбор реализует n турниров, чтобы выбрать n особей. Каждый турнир построен на выборке k элементов из популяции, и выбора лучшей особи среди них. Наиболее распространен турнирный отбор с $k = 2$. Элитные методы отбора гарантируют, что при отборе обязательно будут выживать лучшие члены популяции [9]. Двухточечный и равномерный кроссовер могут использоваться вместо одноточечного оператора. В двухточечном кроссовере выбираются две точки разрыва, и родительские хромосомы обмениваются сегментом, который находится между двумя этими точками. В равномерном кроссовере, каждый бит первого родителя наследуется первым потомком с заданной вероятностью; в противном случае этот бит передается второму потомку.

Гибридные системы

Нечеткие нейронные сети

Преимущества аппарата нечетких нейронных сетей Эффективность аппарата нейросетей определяется их аппроксимирующей способностью, причем НС являются универсальными функциональными аппроксиматорами. 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 рассуждений. 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:

$$R_i: \text{ ЕСЛИ } \ x_{1i} \text{ is } A_{1i} \ \text{ И } x_{2i} \text{ is } A_{2i} \ \text{ ТО } \ z_i \text{ is } C_i.$$

Для реализации базы нечетких правил будем интерпретировать ее как таблицу определения некоторой функции, т. е. базу правил можно представить обучающей выборкой: $\big\{((A_{1i}, A_{2i}), C_i)\big\}$. Например, $\big\{$((малое, большое), около нуля)$\big\}$. 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.3).

При использовании α–срезов каждое α_j–подмножество представляется двумя числами — левой и правой границами: $\alpha^{L_{ij}}$, $\alpha^{R_{ij}}$, где j — номер

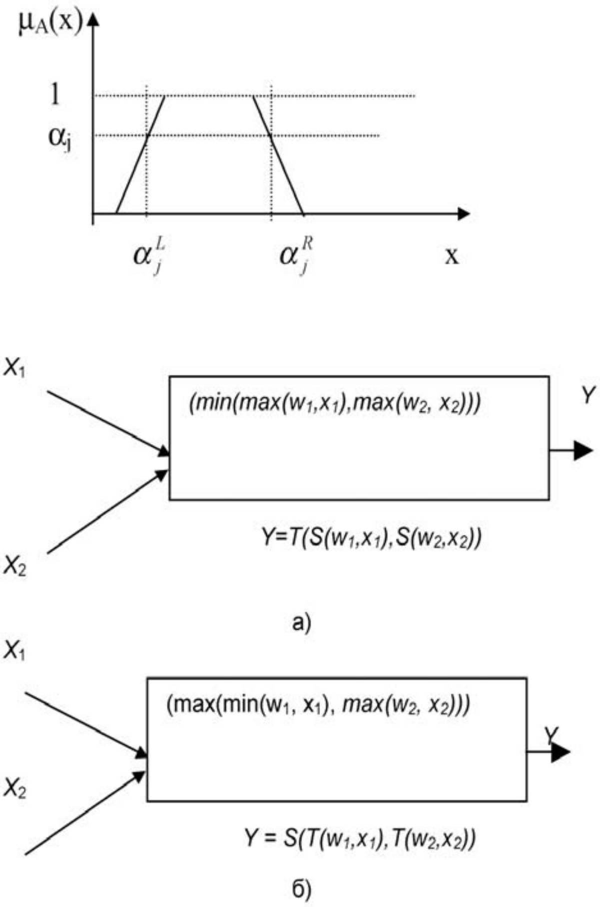


Рис. 3. α–срез, *И-ИЛИ* нейроны

α–среза, а i — номера точек на его левой и правой границах, т. е. α–срезы четко представляют непрерывную функцию принадлежности.

Понятие нечеткой нейросети. Глубинная интеграция нечетких систем и нейросетей связана с разработкой моделей нейронов, функции которых отличаются от функций традиционного нейрона. Модификация модели нейрона для адаптации к нечетким системам касается выбора функции активации, реализации операций сложения и умножения, так как в нечеткой логике сложение моделируется любой треугольной конормой (например, $\max$, $a + b - a * b$, $\ldots$), а операция умножения — треугольной нормой $(\min, a^*b, \ldots)$.

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 S–нормы — $\max$, 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 $F(x) = \exp(-b * (x^2 - a))$.

Нечеткой нейронной сетью (ННС) называют четкую нейронную сеть прямого распространения сигнала, которая построена на основе многослойной архитектуры с использованием *И*-*ИЛИ*-нейронов [11,13,15]. Нечеткая нейросеть функционирует стандартным образом на основе четких действительных чисел. Нечеткой является только интерпретация результатов. При создании гибридной технологии можно использовать нейрокомпьютинг для решения частной задачи нечетких экспертных систем, а именно настройки параметров функции принадлежности. Традиционно функции принадлежности формируют двумя способами: методом экспертной оценки или на основе статистики. Гибридные технологии предлагают третий способ: в качестве функции принадлежности выбирается параметризованная функция формы (например, параметризованная гауссова кривая), параметры которой настраиваются с помощью нейросетей. Настройка параметров может быть получена с помощью алгоритма обратного распространения ошибки.

Рассмотрим применение алгоритма обратного распространения ошибки для обучения ННС. Пусть задана следующая система нечетких правил:

$$\text{ЕСЛИ } x_1 \text{ is } A_{1j} \text{ И } \dots x_n \text{ is } A_{nj} \text{ ТО } z_j \text{ is } C_j.$$

где A_{ij} — нечеткие числа; C_j — действительные числа. Значение $\alpha_j = \Pi_i a_{ji}$ — сила или достоверность правила, α_{ji} — степень принадлежности нечеткого множества условной части j-го правила

$$x_1 \text{ is } A_{1i} \text{ И } \dots x_n \text{ is } A_{ni},$$

$i = 1, \dots, n$ — номер входной переменной, $j = 1, \dots, m$ — количество правил. Значение выхода вычисляется по формуле средневзвешенного выхода:

$$z = \frac{\sum_{j=1}^{m} \alpha_j * z_j}{\sum_{j=1}^{m} \alpha_j}.$$

Допустим, что разработана нейросеть с n входами и одним выходом. Каким образом такая НС может аппроксимировать базу нечетких правил? Любая совокупность нечетких продукций может рассматриваться как нелинейное соответствие, заданное таблицей определения $\{(x_k, y_k)\}$, где $k = 1, \dots, K$ — номер строки-образца в обучающей выборке, x — вектор входа, y — желаемое значение выхода, а z — значение выхода, вычисляемое нейросетью. Если определить текущую ошибку с помощью формулы $E_k = 1/2 * (z_k - y_k)^2$, то можно применить стандартный алгоритм коррекции ошибки, корректируя выход Z по следующему правилу:

$$Z(t+1) = Z(t) - \eta * \left(\frac{\partial E_k}{\partial Z}\right),$$

где η — уровень обучения, $(\partial E_k / \partial Z)$ — направление градиента снижения ошибки. Подставляя в правило формулу для средневзвешенного выхода ННС, получим:

$$Z(t+1) = Z(t) - \eta * (Z_k - Y_k) * \left[\frac{\alpha_j}{a_1 + a_2 + \dots + a_m}\right].$$

При применении стандартного алгоритма обратного распространения ошибки для настройки выхода ННС, необходимо изменить параметры функций принадлежности условных частей правил, т. е. обучение сети позволит их настроить на обучающую выборку.

Структуры гибридных систем (ГС)

Рассмотрим структуры ГС, решающих задачу управления, выделим особенности архитектуры и алгоритмов обучения для каждого конкретного типа ГС [7, 13, 15].

NNFLC — нечеткий контроллер на основе НС. Структура NNFLC приведена на рис. 4.

Структурно NNFLC — это многослойная сеть прямого распространения сигнала, причем различные слои выполняют разные функции. Опишем кратко функции слоев.

- **Слой 1:**
$$y_i^{(1)}(x) = \exp[-(x_i - c_i)^2/2 * \sigma_i^2].$$
Слой 1 представляет функции принадлежности, реализованные как радиальные базисные нейроны.

- **Слой 2:**
$$y_i^{(2)} = \min[y_1^{(1)}, \ldots, y_n^{(1)}].$$
Слой 2 моделирует *И*–условия правил.

- **Слой 3:**
$$y_i^{(3)} = \max[y_1^{(2)}, \ldots, y_n^{(2)}].$$
Слой 3 представляет собой *ИЛИ*–комбинацию правил с одинаковыми термами в консеквентах. Слой 3 выполняет разные функции в рабочем режиме и в режиме обучения. В режиме обучения слой настраивает параметры функций принадлежности выходных переменных. В рабочем режиме формирует значение выхода.

- **Слой 4:**
В рабочем режиме нейроны выполняют дефазификацию:
$$z_i^{(4)} = \sum_j w_{ji} * y_i^{(3)}$$

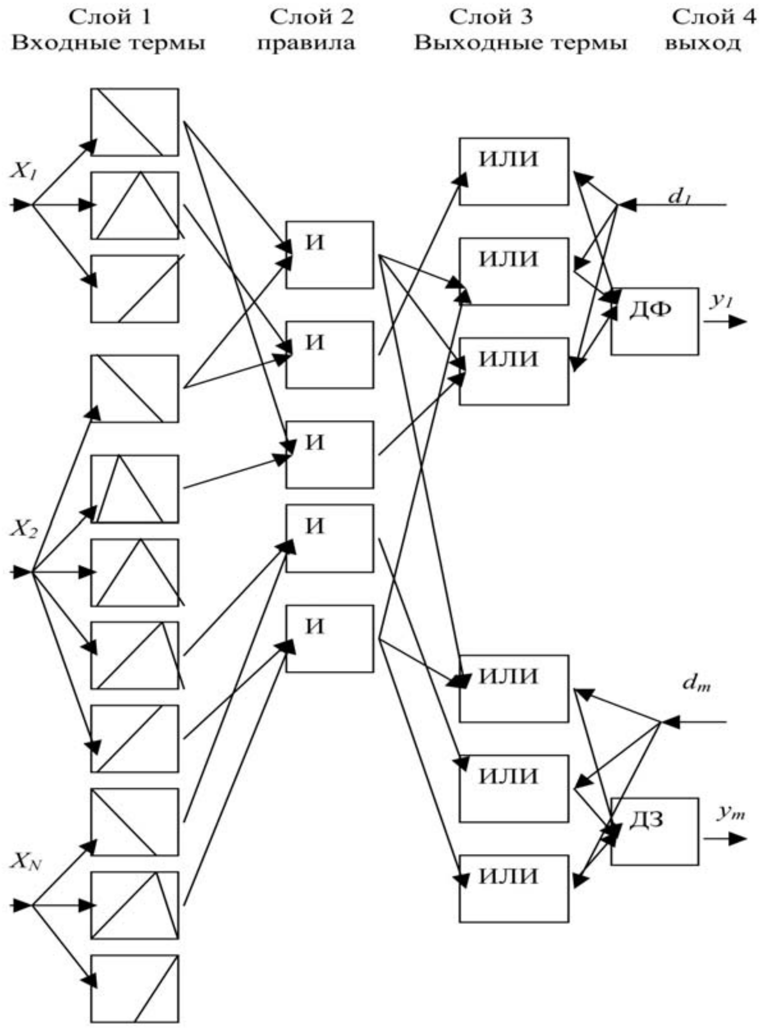


Рис. 4. Структура NNFLC

и нормализацию:

$$y^{(4)}(z_i^{(4)}) = \sum_j y_i^{(3)} * \frac{w_{ji}}{\sum_j w_{ji}},$$

а в режиме обучения — это дополнительный вход, позволяющий настроить функцию принадлежности выходной переменной. Структура ННС NNFLC инициализируется по принципу формирования полной матрицы правил. Если x_i — входные переменные, $\tau(x_i)$ — количество нечетких меток (разбиений) x_i, то исходное количество правил:

$$T = \Pi_{i=1}^{n} \tau(x_i).$$

Обучение ННС сложной архитектуры (с различными функциональными слоями) обычно происходит многоэтапно, причем, на каждом этапе используются различные алгоритмы обучения: предобучение (offline), оперативное (online), без учителя, с учителем. Общая схема обучения ННС NNFLC содержит следующие этапы:

- формирование обучающих данных;
- самоорганизующаяся кластеризация (настройка функций принадлежности);
- соревновательное обучение (алгоритм победителя);
- удаление правил;
- комбинирование правил;
- окончательная настройка параметров (тюнинг) функций принадлежности с помощью алгоритма обратного распространения ошибки.

Приведем содержательные характеристики этапов обучения. Настройка параметров функций принадлежности включает в себя определение центров c_i и ширины σ_i для функций принадлежности, представленных функциями формы:

$$y_i^{(1)}(x) = \exp[-(x_i - c_i)^2/2 * \sigma_i^2].$$

Алгоритм победителя выявляет c_i : $\Delta c_w(t) = \eta(t)(x - c_w(t))$, где $\eta(t)$ — монотонно убывающий уровень обучения. Настройка ширины σ_i осуществляется эвристически, например, по принципу «первого ближайшего соседа»: $\sigma_i = (\sigma_i - \sigma_w)/\lambda$, λ — параметр перекрытия. Алгоритм победителя ищет матрицу весов w_{ji}, которая оценивает качество связей левой и правой частей правил:

$$\Delta w = y_j^{(3)} * (y_i^{(2)} - w_{ji}).$$

Комбинирование правил часто целесообразно выполнять с участием эксперта. Окончательная настройка функций принадлежности выполняется с помощью алгоритма обратного распространения ошибки для функции ошибки $e_i = (y_i^{(4)} - d_k)^2$. Цепочка правил распространяет ошибку до слоя 1 с обратным роутингом. Таким образом, можно сделать вывод о том, что архитектура NNFLC может быть проинтерпретирована как система нечеткого вывода Такаджи-Суджено.

ANFIS — адаптивная НС, основанная на системе нечеткого вывода. Приведем на рис. 5 структуру ANFIS (Adaptive Network Based Fuzzy Inference System) — адаптивной НС для двух правил:

$$\text{ЕСЛИ } x_1 = A_1 \text{ И } x_2 = B_1 \text{ ТО } y_1 = c_{11} * x_1 + c_{12} * x_2$$
$$\text{ЕСЛИ } x_1 = A_2 \text{ И } x_2 = B_2 \text{ ТО } y_2 = c_{21} * x_1 + c_{22} * x_2.$$

Выход ННС формируется по формуле:

$$y = \frac{(w_1 * y_1 + w_2 * y_2)}{(w_1 + w_2)}$$

Слои 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 дефазификацию:

$$y = w_1 * y_1 + w_2 * y_2.$$

Гибридная сеть архитектуры ANFIS обучается с помощью алгоритма обратного распространения ошибки.

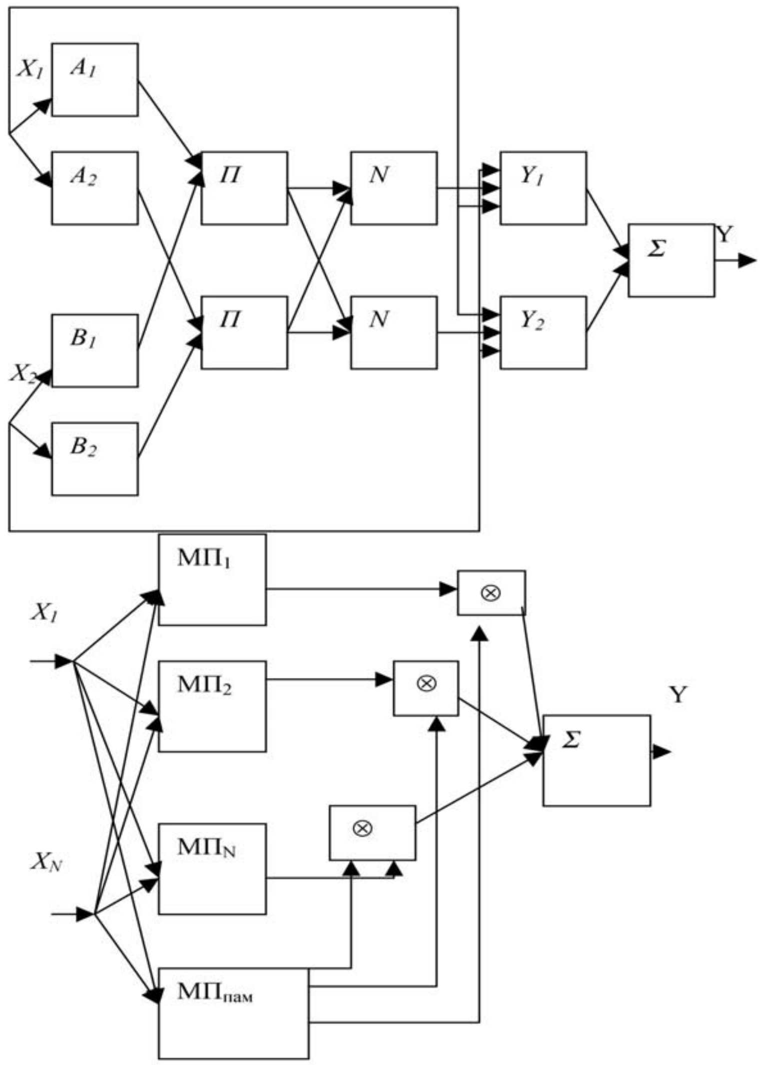


Рис. 5. Структуры ANFIS и NNDFR

NNDFR — НС для нечетких умозаключений. ННС NNDFR (Neuron Network Driven Fuzzy Reasoning) (рис. 5) способна выполнять кластеризацию в пространстве, содержащем нелинейные границы кластеров.

Обозначим через MP_i двухслойный перцептрон с одним скрытым слоем, который служит для описания желаемой кривой поверхности решений, т. е. MP_i выражает правило:

$$\text{ЕСЛИ } x_1 = A_1 \text{ И } x_{ni} = A_{ni} \text{ ТО } y_i = f_i(xi).$$

Для обучения NNDFR предложен алгоритм «удаления правил назад». Алгоритм включает в себя следующие шаги.

- **Шаг 1.** Удаление малозначимых входных переменных. Обучаются отдельные MP_i. После отдельного обучения многослойные перцептроны объединяются и обучаются вместе. Если изменение некоторой входной переменной не влияет на сумму квадратов ошибок, то соответствующая переменная удаляется.
- **Шаг 2.** Формирование обучающей и тестовой выборок. Обучающая выборка $\{x_k, d_k\}$ делится на обучающие и контрольные образцы — паттерны.
- **Шаг 3.** Обучение «многослойного перцептрона (МП) памяти». Эвристически определяется количество кластеров (правил). МП памяти обучается выделять кластеры X. Если x_k принадлежит к A_j, то выход МП памяти $w_j(x_k) = 1$, иначе выход равен 0.
- **Шаг 4.** Обучение MP_i. Многослойные перцептроны MP_i обучаются отдельно методом обратного распространения ошибки.
- **Шаг 5.** Обратное исключение. Используется обратное исключение переменных для каждого MP_i. Общая схема рабочего режима ННС подчиняется следующим правилам:

$$\begin{aligned}
&\text{ЕСЛИ } X = A_1 \text{ ТО } MP_1(x_1, \ldots, x_n),\\
&\text{ЕСЛИ } X = A_2 \text{ ТО } MP_2(x_1, \ldots, x_n),\\
&\text{ЕСЛИ } X = A_n \text{ ТО } MP_{mem}(x_1, \ldots, x_n).
\end{aligned}$$

GARIC — интеллектуальное управление, основанное на обобщенном приближенном выводе. Архитектура GARIC (Generalized Approximate Reasoning Based Intelligent Control) имеет в своем составе две НС различного функционального назначения: НС оценки состояния и НС выбора действия. Используемые алгоритмы обучения — алгоритм усиления и обратного распространения ошибки.

Рассмотрим место GARIC в контуре управления (рис. 6). Рис. 6 и рис. 7 отражают структуру НС управления и НС оценки.

ННС GARIC реализует схему вывода Цукамото. НС управления обучается с помощью алгоритма обратного распространения ошибки. Сложнее обучить НС оценки, которая тренируется алгоритмом усиления:

$$
\begin{gathered}
y_j^n(t+1) = f\Big[\sum_i w_{ij}(t) * x_i(t+1)\Big], \\
f(z) = \big[1 + \exp(-z)\big] - 1, \\
\sigma(r, r_p) = r - r_p, \\
\sigma(r, r_p) \approx r - r_p + \gamma * r_p, \\
0 \leqslant \gamma \leqslant 1,
\end{gathered}
$$

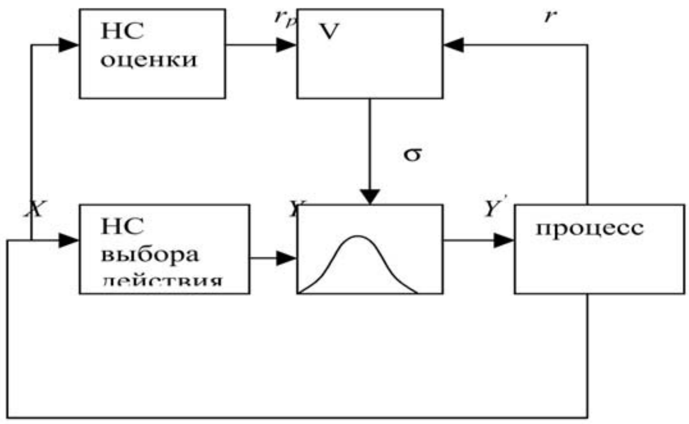
где $\sigma(r, r_p)$ — разница реального и «внутреннего» r. Случайная модуляция y выполняется колоколообразной, случайно распределенной функцией с центром y и разбросом $\sigma(r, r_p)$, что необходимо для покрытия входного пространства. НС оценки представляет собой линейный многослойный перцептрон. Матрицы B и C изменяются стратегией «награда–штраф».

$$
\begin{gathered}
\Delta b_i(t) = \eta * \sigma^{(t+1)} * x_i^t, \\
\Delta c_j(t) = \eta * \sigma^{(t+1)} * y_j^t.
\end{gathered}
$$

Матрица W обучается упрощенным алгоритмом обратного распространения ошибки:

$$
\Delta w_{ij}^{(t)} = \eta * \sigma^{(t+1)} * \mathrm{sign}[c_j^{(t)}] * y_j^t * (1 - y_j^{(t)}) * x_i^{(t)}.
$$

Нечеткая сеть Fuzzy Net (FUN) Нечеткая нейронная сеть FUN (рис. 7) предложена для управления перемещениями мобильного робота. В этой сети осуществляется структурное и параметрическое обучение — последовательный стохастический поиск в пространствах правил и входов. Обучение



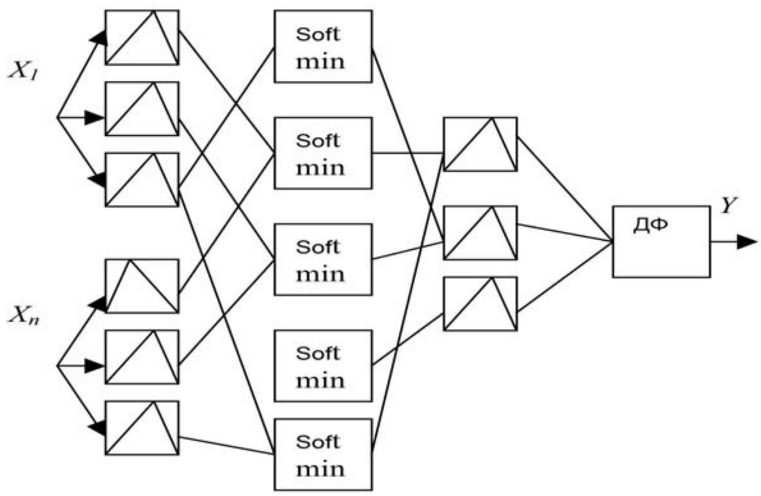


Рис. 6. Нейро-нечеткий контроллер GARIC.НС управления

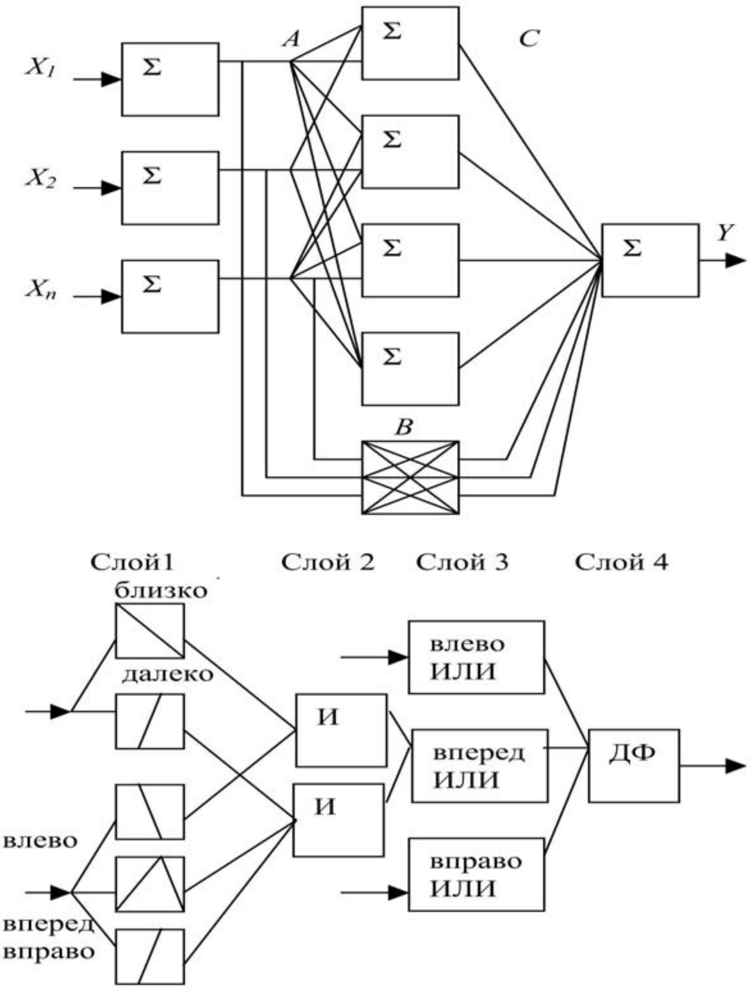


Рис. 7. Структура НС оценки GARIC

правил построено следующим образом. Случайно выбранная связь правил до и после слоя 2 изменяется и проверяется сеть на улучшение ценовой функции. Плохие изменения отменяются, а хорошие сохраняются. Параметры функций принадлежности итерационно изменяются от некоторого случайного значения: $D_p^{(t)} - D_p(t-1) = -\eta * D_p^{(t-1)}$, где $\eta = 10^{-3} \ldots 10^{-2}$ — фактор сходимости.

Таким образом, анализ конкретных архитектур нечеткой нейронной сети позволяет сделать следующий вывод. Наряду с классическими нейронами, являющимися пороговыми суммирующими элементами, должна включать в себя «логические» нейроны, моделирующие логические связки. Фундаментальная задача объединения методов перцепции (восприятия) и логического (абстрактного) мышления не могла не отразиться на уровне структуры ГС. В нечеткой нейронной сети любой из известных архитектур, слои нейронов становятся специализированными. Существуют слои, выполняющие распознавание, и слои, вычисляющие логические формулы и импликации (правила ЕСЛИ . . . ТО). Причем конкретная схема нечеткой нейронной сети зависит от решаемой задачи.

Нечеткий нейронный контроллер. Рассмотрим особенности применения нечетких нейронных сетей на примере нечеткого регулятора для стиральной машины, то есть построим теперь fuzzy-neuro контроллер. Выберем для демонстрации технологии нечетких нейронных сетей архитектуру ANFIS. Основа интеграции нейронных сетей и систем нечеткого вывода заключается в том, что оба метода представляют нелинейное отношение в пространстве входов и выходов. Важная задача нечеткого моделирования — это настройка функций принадлежности, являющаяся по существу задачей оптимизации. Для ее решения используются как нейронные сети, так и генетические алгоритмы. Самый простой подход к настройке функций принадлежности заключается в том, что выбирают определенную параметризованную функцию формы и подбирают ее параметры на основе обучения нейронной сети. Рассмотрим простой контроллер с тремя нечеткими правилами вывода в базе знаний:

R_1 : ЕСЛИ «количество белья» = «много»
И «температура воды» = «высокая»
И «загрязненность» = «высокая»
ТО «длительность» = «высокая» ;
R_2 : ЕСЛИ «количество белья» = «много»
И «температура воды» = «высокая»
И «загрязненность» = «низкая»
ТО «длительность» = «низкая» ;
R_1 : ЕСЛИ «количество белья» = «мало»
И «температура воды» = «низкая»
И «загрязненность» = «низкая»
ТО «длительность» = «низкая» .

Зададим следующие функции формы для правил. Пусть выражению «количество белья» = «мало» соответствует функция $L_1(x)$, а «количество белья» = «много» функция $H_1(x)$:

$$L_1(x) = \frac{1}{(1+\exp(b_1(x-c_1))},$$
$$H_1(x) = \frac{1}{(1+\exp(-b_1(x-c_1))},$$
$$L_1(x) + H_1(x) = 1.$$

Аналогично для выражения «температура воды» = «высокая» будем использовать функцию $L_2(t)$, а для «температура воды» = «низкая» функцию $H_2(t)$:

$$L_2(t) = \frac{1}{(1+\exp(b_2(t-c_2))},$$
$$H_2(t) = \frac{1}{(1+\exp(-b_2(t-c_2))},$$
$$L_2(t) + H_2(t) = 1.$$

Для выражения «загрязненность» = «высокая» определим функцию $L_3(z)$ и для «загрязненность» = «низкая» функцию $H_3(z)$:

$$L_3(z) = \frac{1}{(1+\exp(b_3(z-c_3))},$$
$$H_3(z) = \frac{1}{(1+\exp(-b3(z-c3))},$$
$$L_3(z) + H_3(z) = 1.$$

Для выходов «длительность» = «высокая» и «длительность» = «малая» аналогично определим функции

$$L_4(y) = \frac{1}{(1+\exp(b_4(y-c_4))},$$
$$H_4(y) = \frac{1}{(1+\exp(-b_4(y-c_4))},$$
$$L_4(x) + H_4(x) = 1.$$

Система нечеткого вывода по Цукамото представлена на рис. 8.

Для четких значений «количество белья», «температура воды» и «загрязненность»: A_1, A_2, A_3 определим релевантность (силу) правил α_i, как показано на рис. 8:

$$\alpha_1 = H_1 \cap H_2 \cap H_3,$$
$$\alpha_2 = H_1 \cap H_2 \cap L_3,$$
$$\alpha_3 = L_1 \cap L_2 \cap L_3.$$

Выходы по каждому из правил определяется с помощью обратных функций принадлежности правых частей правил.

$$Y_1 = H_4^{-1}(\alpha_1),$$
$$Y_2 = H_4^{-1}(\alpha_2),$$
$$Y_3 = L_4^{-1}(\alpha_3).$$

Общий выход из системы нечетких правил определяется как

$$y_0 = \frac{(\alpha_1 * Y_1 + \alpha_2 * Y_2 + \alpha_3 * Y_3)}{(\alpha_1 + \alpha_2 + \alpha_3)}.$$

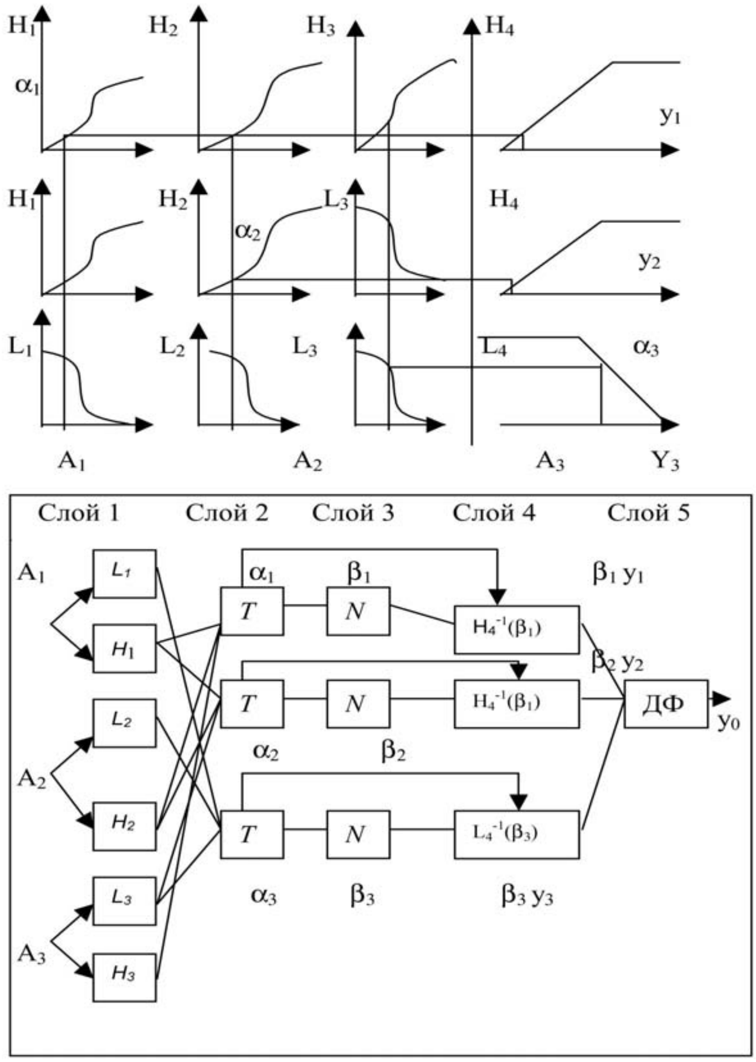


Рис. 8. Нечеткий вывод и структура нейро-нечеткого контроллера

Далее построим нечеткую нейронную сеть, идентичную системе нечеткого вывода и обучим функции принадлежности анцедента и консеквента правил (рис. 8):

- **Слой 1.** Выходы узлов — это степени, с которыми заданные входы удовлетворяют функциям принадлежности, ассоциированным с этими узлами.
- **Слой 2.** Каждый узел вычисляет силу правила. Выход верхнего нейрона $\alpha_1 = H_1 \cap H_2 \cap H_3$, выход среднего нейрона $\alpha = H_1 \cap H_2 \cap L_3$, а выход нижнего: $\alpha_3 = L_1 \cap L_2 \cap L_3$. Все узлы помечены T, так как можно выбрать любую T–норму для моделирования логического *И*. Узлы этого слоя называются узлами правил.
- **Слой 3.** Каждый узел помечен N, чтобы показать, что узлы нормализуют силу правил

$$\beta_i = \frac{\alpha_i}{(\alpha_1 + \alpha_2 + \alpha_3)}.$$

- **Слой 4.** Выход нейронов — это произведение нормализованной силы правила и индивидуального выхода соответствующего правила.

$$Y_1 = H_4^{-1}(\alpha_1),$$
$$Y_2 = H_4^{-1}(\alpha_2),$$
$$Y_2 = L_4^{-1}(\alpha_3).$$

- **Слой 5.** Одиночный выходной нейрон вычисляет выход сети

$$y_0 = \beta_1 * Y_1 + \beta_2 * Y_2 + \beta_3 * Y_2.$$

Алгоритмы обучения для нечеткой нейронной сети контроллера. Пусть задана четкая обучающая выборка

$$\{(x_1, t_1, z_1, y_1), \ldots, (x_k, t_k, z_k, y_k)\},$$

где x, t, z — входные условия «количество белья», «температура воды» и «загрязненность», а y — длительность стирки. Ошибку на k-ом образце определим как обычно $E^k = \frac{1}{2} * (O_i - Y_i)^2)$. Можно настроить параметры функций принадлежности в ходе обучения нечеткой нейронной сети, используя следующие соотношения как правила изменения весов в алгоритме обратного распространения ошибки:

$$b_4(t+1) = b_4(t) - \eta * \frac{\partial E}{\partial b_4},$$
$$c_4(t+1) = c_4(t) - \eta * \frac{\partial E}{\partial c_4},$$
$$\ldots$$
$$c_1(t+1) = c_1(t) - \eta * \frac{\partial E}{\partial c_1}.$$

Нечеткие нейронные сети с генетической настройкой Настройка функций принадлежности с помощью нейронной сети или ГА устраняет принципиальную слабость теории нечетких систем — субъективность функций принадлежности. Если настроить функцию принадлежности с помощью нейронной сети, то окончательная форма функции будет аппроксимацией обучающей выборки. С помощью ГА, т. е. стохастической оптимизации также можно настроить функции принадлежности. Используем традиционный градиентный метод для обучения параметров левой и правой частей нечетких правил. Покажем, как можно настроить параметры функции формы.

Генетической нечеткой системой называют нечеткую систему, функции принадлежности и база правил которой спроектирована с помощью генетического алгоритма. Для применения ГА необходимо выполнить следующие действия:

- выбрать единицу кодирования, т. е. хромосому;
- уточнить эволюционные операторы рекомбинации, мутации и селекции;
- сформировать функцию оптимальности (fitness function, performance index).

В настоящее время ГА используют либо для настройки функций принадлежности (базы данных), либо для формирования базы правил (базы знаний), либо для одновременного формирования и функций принадлежности, и правил. В соответствии с объектами оптимизации выбирают единицу кодирования — хромосому. Для настройки функций принадлежности за хромосому выбирают одно правило (мичиганский подход), для настройки базы правил за хромосому выбирают вариант базы правил (питтсбургский подход, подход итеративного обучения правил) [12, 17]. В соответствии с кодированием уточняются правила генерации новых хромосом.

Функция оптимальности представляет собой механизм нечеткого вывода, который для каждого варианта базы правил строит либо управляющее воздействие для нечеткого контроллера, либо экспертное заключение для диагностической экспертизы. Как видно из описания механизма нечеткого вывода, все нечеткие правила вносят вклад в окончательный результат, то есть правила сотрудничают. Но при отборе правил (хромосом) ГА сохраняет только правила, внесшие максимальный вклад в общий результат, т. е. хромосомы конкурируют. Говорят о проблеме «конкуренции и кооперации» в генетических нечетких системах (a competition vs. a cooperation). Решение проблемы в каждом конкретном случае строится эвристически, например, при подходе итеративного обучения правил используют два этапа оптимизации. На первом шаге правила конкурируют за право войти в базу правил, а на втором — взаимодействуют при формировании общего результата.

Системы генетического проектирования нечетких нейронных сетей

Рассмотрим возможности генетических вычислений, как средства структурной оптимизации нечетких нейронных сетей. Генетические вычисления можно применить на этапе проектирования нейронной сети, как предиктора временных рядов, в том числе можно проектировать и нечеткую нейронную сеть. Возможности нейронных сетей интерполировать значения временных рядов широко используют для оценки поведения макроэкономических показателей, в том числе индексов деловой активности или уровня ценных бумаг. Задача построения нейронного предиктора связана с принятием решений на разных этапах проектирования. Необходимо исследовать входные данные и решить, по какому отрезку входных данных рационально делать предсказания, сколько точек в будущем разумно предсказать. Пространство перебора решений огромно для развитых рынков ценных бумаг, накопивших статистические сведения за десятки лет. Необходимо принять решение и о типе нейронной сети: многослойный персептрон, радиально базисная сеть, вероятностная нейронная сеть, сеть регрессии и т. д. Для выбранного типа НС необходимо определить количество скрытых слоев, количество нейронов в них, виды функций активации и другие. Для каждой сети необходимо выбрать алгоритм обучения и его параметры, например, использование моментов (уровень моментов), уровень обученности, целевой уровень накопленной ошибки и т.д. Для нечеткой нейронной сети необходимо определить архитектуру сети, параметры функций принадлежности. В результате пространство решений при формировании нечеткого нейронного предиктора становится необозримым для исследователя и целесообразно применить ГА как средство эволюционного проектирования.

Примером такого средства может служить программная система NeuroForecaster GENETICA (NFGA Copyright 1993–1998, NIBS Pte Ltd, 62 Fowlie Road, Republic of Singapore 428501.) Программа NFGA — это программа поиска и оптимизации НС, основанная на генетических алгоритмах, созданная для поиска лучшей комбинации входных данных, лучшего периода прогноза (т. е. количества прогнозируемых значений) для данной проблемы.

Заключение

В настоящее время рассмотренные технологии «вычислительного интеллекта» являются успешными прикладными технологиями. Самыми яркими событиями были [1,4] запуск в 1987 году системы управления новым метро в г. Сендай около Токио, достижение японского экспорта изделий с fuzzy logic к 1991 году цифры в 25 млрд. долл. США. Это, в первую очередь, товары культурно-бытового назначения — фотоаппараты, видеокамеры, стиральные машины, холодильники, пылесосы, микроволновые печи и многое другое. Международный научно-исследовательский институт (LIFE — Laboratory for International Fuzzy Engineering Research), созданный в Японии располагал в 1993 году бюджетом в 64 млн. долл. Таким образом, «вычислительный интеллект» — это успешная электронная и программная индустрия.

Литература

1. *Абдусаматов Р. М., Беркинблит М. Б., Фельдман А. Г., Чернавский А. В.* Моторика и интеллект // Интеллектуальные процессы и их моделирование. – М.: Наука, 1987.

2. *Аверкин А. Н. и др.* Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. *Д. А. Поспелова*. – М.: Наука, 1986.

3. *Аверкин А. Н., Федосеева И. Н.* Параметрические логики в интеллектуальных системах управления. – М.: Вычислительный центр РАН, 2000.
4. *Васильев В. И., Ильясов Б. Г.* Интеллектуальные системы управления с использованием генетических алгоритмов. Учебное пособие. – Уфа: УГАТУ, 1999.
5. *Заде Л. А.* Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976.
6. *Клир Дж.* Системология. Автоматизация решения системных задач: Пер. с англ. – М.: Радио и связь, 1990.
7. *Кофман А.* Введение в теорию нечетких множеств. – М.: Радио и связь, 1982.
8. *Мелихов А. Н., Берштейн Л. С., Коровин С. Я.* Ситуационные советующие системы с нечеткой логикой. – М.: Наука, 1990.
9. *Наместников А. М., Ярушкина Н. Г.* Эффективность генетических алгоритмов для задач автоматизированного проектирования //Известия РАН. Теория и системы управления. – 2002. – № 2.
10. *Скурихин А. Н.* Генетические алгоритмы // Новости искусственного интеллекта. – 1995. – № 4.
11. *Ярушкина Н. Г.* Нечеткие нейронные сети // Новости искусственного интеллекта. – 2001. – № 2–3.
12. *Ярушкина Н. Г.* Гибридные системы, основанные на мягких вычислениях: определение, архитектура, возможности // Программные продукты и системы. – 2002. – № 3.
13. *Bothe H.-H.* Fuzzy Neural Networks. – Prague: IFSA, 1997.
14. *Deichelmann H.*. Linguistishe Systeme und ihre Anwendung. – Darmstadt: Fachhochschule Darmstadt, 1996.
15. *Fuller R.* Fuzzy systems.
URL: http://www.abo.fi/~rfuller/
16. *Goldberg D. E.* Genetic Algorithms in Search, Optimization and Machine Learning. – Adison–Wesley Publishing Company Inc., 1989.
17. *Herrera F., Magdalena L.* Genetic fuzzy systems. – Prague: IFSA, 1997
18. *Holland Y.* Adaptation in Natural and Artificial Systems. – University of Michigan, Press, Ann Arbor,1975.
19. *Schweizer B., Sklar A.* Associative functions and abstract semigroups // Publ. Math., No. 10, 1963.
20. *Zadeh L. A.* Fuzzy Sets // Information and Control, **8** (1965), 338–353.
21. *Zadeh L.* What is soft computing? // Soft Computing, 1997, **1**, pp.1–2.

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 150 научных публикаций.

НАУЧНАЯ СЕССИЯ МИФИ–2004

НЕЙРОИНФОРМАТИКА–2004

**VI ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ
ПО НЕЙРОИНФОРМАТИКЕ

Часть 1

Оригинал-макет подготовлен *Ю. В. Тюменцевым*
с использованием издательского пакета LaTeX 2_ε
и набора PostScript–шрифтов *PSCyr*

Подписано в печать 25.11.2003 г. Формат 60 × 84 1/16
Печ. л. 12, 5. Тираж 200 экз. Заказ №

Московский инженерно-физический институт
(государственный университет)
Типография МИФИ
115409, Москва, Каширское шоссе, 31