

УДК 519.71

О. Г. Руденко, А. В. Островерхий, Н. Н. Островерхая  
**АППРОКСИМАЦИЯ МНОГОМЕРНЫХ ФУНКЦИЙ  
 С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ СМАС**

**1. Введение**

Искусственная нейронная сеть (ИНС) СМАС, предложенная Дж. Альбусом [1] для управления роботом-манипулятором, хорошо зарекомендовала себя при решении задач управления, идентификации, кодирования, обработки и сжатия изображений, т. е. задач, в которых используется аппроксимация каких-либо функций. В данных задачах размерность рассматриваемых пространств обычно не превышает трех. В свою очередь задача аппроксимации существенно усложняется с увеличением размерности решаемой задачи.

Целью данной работы является изучение различных архитектур СМАС, позволяющих эффективно решить задачу аппроксимации нелинейных функций большой размерности.

**2. Архитектура сети СМАС**

Сеть СМАС, пример которой приведен на рис. 1, в общем случае осуществляет следующие преобразования:

$$S: X \Rightarrow A, \tag{1}$$

$$P: A \Rightarrow y. \tag{2}$$

где  $X$  —  $N$ -мерное пространство непрерывных входных сигналов;  $A$  —  $n$ -мерное пространство ассоциаций;  $y$  — вектор выходных сигналов.

Преобразование (1) соответствует кодированию информации

$$a = S(x), \tag{3}$$

а преобразование (2) — вычислению выходного сигнала.

Входной слой состоит из нейронов (возбужденные нейроны отмечены на рис. 1 темным цветом).

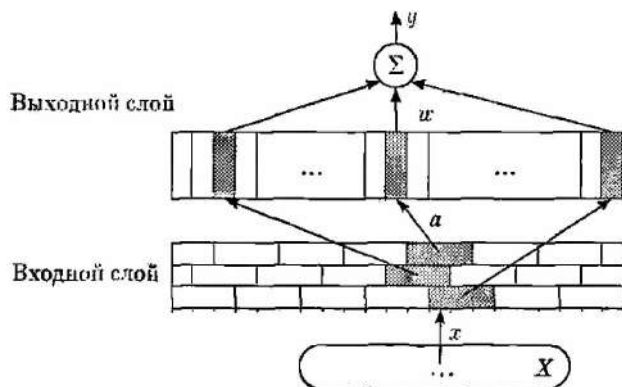


Рис. 1. Структура сети СМАС

имеющих, как правило, одинаковые базисные (активационные) функции (БФ). В традиционной сети СМАС БФ выбраны прямоугольными с единичной амплитудой, поэтому в данной сети преобразование (2) осуществляется в соответствии с формулой:

$$\hat{y} = P(a) = a^T w = S(x)^T w. \tag{4}$$

Таким образом, сеть СМАС работает по принципу ассоциативной памяти. Ассоциативные свойства СМАС проявляются в используемом виде адресации, основанном на специальном кодировании входной информации. Максимальное количество ассоциативных нейронов — ячеек памяти для хранения весовой информации — можно определить по формуле:

$$n_{max} = \left\lfloor \rho \left( \frac{R-1}{\rho} + 1 \right)^N \right\rfloor. \tag{5}$$

где  $R$  — используемое число уровней квантования входных сигналов;  $\rho$  — число ступеней квантования;  $N$  — размерность входного вектора  $x$ ;  $\lfloor \bullet \rfloor$  означает округление в сторону ближайшего большего целого числа.

Обучение сети СМАС, как практически и всех других ИНС, заключается в выстройке вектора ее весовых параметров  $w$  размерности  $n \times 1$ . В традиционной СМАС [1, 2] на каждом такте обучения сети предъявляются обучающие пары  $\{x(k), y(k)\}$ , где  $y(k)$  — значение функции, соответствующее  $x(k)$ , и корректируются лишь те ее  $\rho$  весов, которые соответствуют единичным компонентам вектора ассоциаций для данного вектора  $x(k)$ . При этом правило обучения для всех  $i, j$ , для которых  $a_i(k) = a_j(k) = 1$ , имеет вид:

$$w_j(k+1) = w_j(k) + \gamma \cdot \left( y(k) - \frac{1}{\rho} \sum_{i=1}^{\rho} w_i(k) \right), \tag{6}$$

где  $\gamma \in (0, 1]$  — параметр, влияющий на скорость обучения.

**3. Выбор базисных функций**

Выбор БФ играет важную роль в формировании аппроксимирующих свойств сети СМАС. Как было отмечено выше, традиционная сеть СМАС использует БФ прямоугольной формы, однако в некоторых задачах такой выбор является неэффективным, а в ряде случаев и неприемлемым. Если в сети используются нейроны с базисными функциями, отличными от прямоугольной, то преобразование (4) принимает вид

$$\hat{y} = a^T \Phi(x)w, \quad (7)$$

$$\text{где } \Phi(x) = \begin{bmatrix} \Phi_1(x) & 0 & \dots & 0 \\ 0 & \Phi_2(x) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \Phi_n(x) \end{bmatrix},$$

$\Phi_i(x) = \prod_{j=1}^N \phi_{ij}(x_j)$ ,  $\phi_{ij}(x_j)$  — значение выбранной базисной функции в точке  $x_j$ .

При выборе БФ прямоугольной формы вычислительные затраты будут минимальными, так как во входном слое сети вместо вычисления значений функций активации для каждого возбужденного нейрона достаточно определить какие нейроны возбуждены и подать на их выходы сигнал, равный единице. Скорость обучения сети при выборе базисных функций прямоугольной формы будет максимальной. Существенным также является тот факт, что в сети СМАС с БФ прямоугольной формы схема кодирования информации не влияет на точность аппроксимации, т. е. любая схема кодирования является приемлемой [3]. Однако сети с единичными БФ, помимо невысокой точности осуществляемой кусочно-постоянной аппроксимации, обладают еще одним недостатком: давние сети не применимы в случаях, когда требуется дифференцирование БФ.

Существует целый класс монотонно дифференцируемых БФ, лишенных данного недостатка. Так, в работах [3, 4] предложено в качестве базисных применять гауссовские функции, а в [6] — В-сплайны различных порядков, основным достоинством которых, помимо дифференцируемости на всей области определения, является простота вычисления их производных. Однако в [3] отмечено, что данные БФ имеют вблизи границы области квантования (зоны чувствительности нейрона входного слоя) сравнительно небольшие значения, что негативно сказывается на результатах аппроксимации, и для избежания указанного недостатка в качестве базисных рекомендуется использование тригонометрических функций, например, косинусоидальной

$$\Phi(x) = \begin{cases} \cos\left(\pi \cdot \frac{x-m}{\lambda}\right), & \text{при } x \in \left[m - \frac{\lambda}{2}; m + \frac{\lambda}{2}\right]; \\ 0, & \text{в противном случае,} \end{cases} \quad (8)$$

где  $m$  — центр гиперкуба,  $\lambda$  — длина гиперкуба.

Она также позволяет легко вычислять производные произвольных порядков и равна нулю на границах области определения.

Еще более простыми и в то же время эффективными являются, на наш взгляд, БФ параболической формы. Параболические БФ имеют форму, близкую к тригонометрическим, и позволяют легко вычислять значение производной, но при этом обладают

еще большей крутизной роста вблизи границ области квантования и требуют значительно меньших вычислительных затрат при их реализации. БФ параболической формы и их производная имеют соответственно вид:

$$\Phi(x) = \begin{cases} 1 - \left(2 \cdot \frac{x-m}{\lambda}\right)^2, & \text{при } x \in \left[m - \frac{\lambda}{2}; m + \frac{\lambda}{2}\right]; \\ 0, & \text{в противном случае.} \end{cases} \quad (9)$$

$$\Phi'(x) = -4 \cdot \frac{(x-m)}{\lambda}. \quad (10)$$

Здесь приняты те же обозначения, что и в (8).

Следует отметить, что выбор БФ, отличных от прямоугольных, приводит к изменению алгоритма обучения (6), который в этом случае может быть записан так:

$$w(k+1) = w(k) + \gamma(k) \cdot \left( \frac{y(k) - a^T \Phi(x)w(k)}{\|\Phi(x)a(k)\|^2} \Phi(x)a(k) \right), \quad (11)$$

где  $\gamma(k)$  — векторный, в общем случае переменный, параметр.

Свойства алгоритма (11) в значительной степени зависят от выбора  $\gamma(k)$ . Несложно показать, что оптимальное значение этого параметра, обеспечивающее максимальную скорость обучения при отсутствии помех, будет равно единице. Для обеспечения же сходимости алгоритма (11) при наличии помех измерений параметр  $\gamma(k)$  должен удовлетворять условиям Дворецкого. В частности, достаточно эффективным оказывается выбор  $\gamma(k)$  вида

$$\gamma(k) = \gamma \alpha^k, \quad (12)$$

где  $\gamma(k) \in (0, 1)$ ;  $0 < \alpha \ll 1$ .

#### 4. Проблемы аппроксимации многомерных функций сетью СМАС

Как можно видеть из формулы (5), с ростом размерности пространства входных переменных  $N$  объем памяти, требуемый для хранения информации о весах, растет экспоненциально. Ограниченные размеры памяти сужают сферу применения СМАС в реальных приложениях. Кроме того, увеличение размерности  $N$  приводит к возрастанию сложности кодирования информации.

Одним из известных способов решения данной проблемы является хеширование, заключающееся в отображении большой требуемой памяти в меньшую, физически реализуемую. Однако при этом неизбежно возникает проблема хеш-коллизий, заключающихся в появлении нежелательной корреляции между входными сигналами, весьма отрицательно сказывающихся на аппроксимирующих свойствах сети [7].

В качестве более эффективного способа уменьшения объема требуемой памяти в сети СМАС при работе с многомерными объектами в [4] предлагается построение перархической структуры СМАС (Hierarchical СМАС — НСМАС), состоящей из нескольких более простых модулей, например, двумерных СМАС. На рис. 2 приведена наименьшая топология НСМАС, учитывающая, что каждая СМАС содержит два входа и выходной сигнал СМАС первого слоя является входным сигналом для СМАС второго слоя и т. д. Архитектура НСМАС может быть соответственно расширена с использованием данной топологии бинарного дерева.

На рис. 2 использованы следующие обозначения:  $x_i$  ( $i = 1, 2, \dots, N$ ) —  $i$ -й вход нейронной сети НСМАС; ( $j = 1, 2, \dots, n$ ;  $l = 1, 2, \dots, L$ ) — выход  $j$ -го СМАС слоя  $l$ ;  $n$  — число сетей СМАС в слое  $l$ ;  $y(x)$  — выходной сигнал НСМАС для входного сигнала  $x$ .

Если выбранные БФ дифференцируемы, то для обучения данной сети (настройки ее параметров) может быть применен алгоритм обратного распространения ошибки, когда вначале настраиваются параметры СМАС выходного слоя  $L$ , а затем ошибка распространяется в обратном направлении от  $L$ -го слоя к  $(L - 1)$ -му скрытому слою для настройки параметров СМАС  $(L - 1)$ -го слоя и т. д. При этом в качестве минимизируемого параметра выбирается квадратичный функционал ошибки

$$E = \frac{1}{2}(\hat{y}(x) - y(x))^2, \quad (13)$$

где  $y(x)$ ,  $\hat{y}(x)$  — требуемый и реальный выходные сигналы НСМАС для входного сигнала  $x$  соответственно.

Таким образом, обобщенная процедура настройки сети может быть записана так.

1) *Настройка СМАС выходного слоя* — практически ничем не отличается от настройки обычной

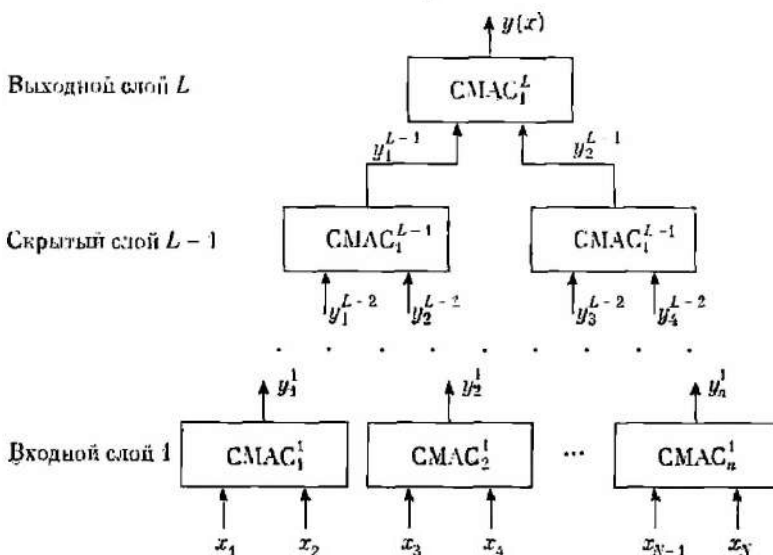


Рис. 2. Топологическая структура нейронной сети НСМАС

двухвходовой СМАС, например по алгоритму (11), за исключением того, что входными для данного слоя являются выходные сигналы предыдущего слоя.

2) *Настройка сетей СМАС скрытых слоев* по правилу

$$w_h^l(k+1) = w_h^l(k) + \gamma \cdot (\hat{y}_{[h/2]}^{l+1}(y^l) - y_{[h/2]}^{l+1}(y^l)) \times \frac{\partial y_{[h/2]}^{l+1}(x)}{\partial y_h^l} \cdot \Phi(y^l) a(k), \quad (14)$$

где

$$\frac{\partial y_{[h/2]}^{l+1}(x)}{\partial y_h^l} = \sum_{i=1}^p a_i(y^l) \cdot w_{hi}^l \cdot \left[ \prod_{j=2}^{[2]^{h/2}+1} \Phi_j(y_j^l) \right] \cdot \frac{\partial \Phi_i(y_h^l)}{\partial y_h^l}, \quad (15)$$

$h = 1, 2, \dots, n$ ;  $w_{hi}^l$  — значение веса в  $i$ -й ячейке памяти СМАС $_h$  слоя  $l$ .

Данная процедура повторяется до тех пор, пока не будет достигнуто требуемое значение критерия ошибки либо заданное максимальное число итераций обучения.

Известные проблемы, сопутствующие реализации алгоритма обратного распространения ошибки, при исследовании многомерных функций могут быть исключены при использовании другого подхода — применения Low-Dimensional-СМАС-Based сетей (LDB СМАС), предложенных в [8]. Пример структуры LDB СМАС сети показан на рис. 3, где  $m$  — число СМАС,  $N$  — число входных переменных.

В основе данной структуры лежит множество малоразмерных (базовых) сетей СМАС (например, двумерных), на которые подаются все возможные парные комбинации входных сигналов. Взвешенные веса  $z_i$  ( $i = 1, 2, \dots, m$ ) выходы этих СМАС формируют общий выход сети

$$y(x) = \sum_{i=1}^m y_i(x) z_i, \quad (16)$$

где  $y_i(x)$  — выходное значение  $i$ -й сети СМАС;  $z_i$  — вес соответствующего выхода  $y_i(x)$  ( $i = 1, 2, \dots, m$ ).

Данная архитектура дает возможность создания и неполных структур, в которых участвуют не все возможные комбинации пар входных переменных, а лишь часть их.

Процедура обучения сети LDB СМАС состоит в следующем:

1) *Настройка весов  $z_i$* , например, по алгоритму

$$z(k+1) = z(k) + \frac{y(x) - (x) Y(k)}{\|Y(k)\|^2} Y(k), \quad (17)$$

где  $Y(k) = (y_1(k), y_2(k), \dots, y_m(k))$  — вектор выходов базовых сетей СМАС.

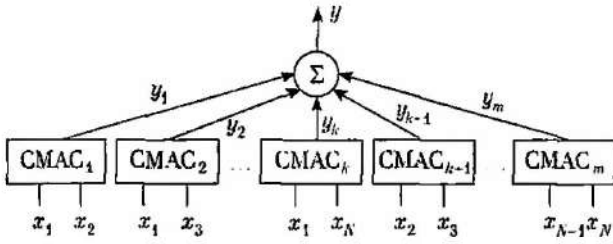


Рис. 3. Структура сети LDB SMAS

2) Настройка параметров базовых сетей  $СМАС_i$  ( $i = 1, 2, \dots, m$ ) по правилу

$$w(k+1) = w(k) + \frac{y(x) - \hat{y}(x)}{\|\Phi(k)a(x)\|^2} \Phi(k)a(x). \quad (18)$$

Следует отметить, что данная структура не является иерархической, что позволяет применять градиентный метод обучения при выборе БФ любой формы, включая прямоугольную.

5. Экспериментальные исследования

Исследования производились на ПК с процессором Intel Pentium 4 3,2 GHz в среде MatLab 7.0 под ОС Linux 2.6.19.

Все последующие базовые сети СМАС использовали  $R = 50$  уровней квантования, распределенных по  $p = 10$  ступеням квантования. Двумерная сеть СМАС с такими характеристиками требует 350 ячеек памяти. Обучение сети осуществлялось на выборке из 20 000 случайных точек, равномерно распределенных на интервале  $[-1; 1]$ .

Рассматривалась задача аппроксимации нелинейной четырехмерной функции  $F(x_1, x_2, x_3, x_4)$

$$F(x_1, x_2, x_3, x_4) = x_1 + \sin(\pi x_1) \cdot \cos(\pi x_2) \times \sin(\pi x_3) \cdot (\sin(\pi x_4)^2 - 1). \quad (19)$$

Сечение данной функции при  $x_2 = x_3 = 0,25$ , т. е.  $G(x_1, x_4) = F(x_1, 0,25, 0,25, x_4)$  показано на рис. 4.

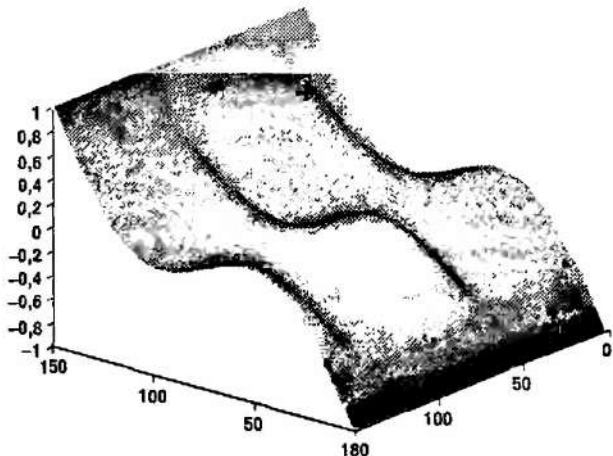


Рис. 4. Вид исходной функции  $F(x_1, 0,25, 0,25, x_4)$

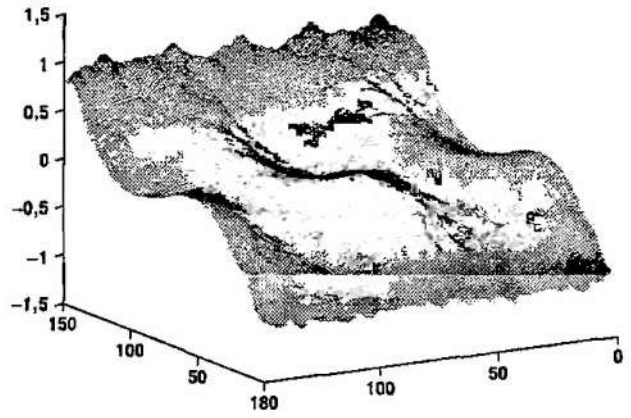


Рис. 5. Аппроксимация сетью СМАС с параболическими БФ без нормализации

Для оценки эффективности работы сетей использовались следующие параметры:

1. Время вычислений в секундах —  $T$ ;
2. Средняя квадратичная ошибка:

$$MSE = \sqrt{\frac{\sum_{i=1}^M (y_i(x) - \hat{y}_i(x))^2}{M}}$$

где  $M$  — количество экспериментов.

Эксперимент 1. Целью данного эксперимента было исследование выбора различных БФ и нормализаций.

Некоторые результаты вычислений при отсутствии и наличии того или иного вида нормализации приведены в табл. 1 и на рис. 5–7. Здесь приняты следующие сокращения: БН — без нормализации.

Таблица 1

Вид базисных функций	БН		НЧП		ПГК		НЧП+ПГК	
	$T, c$	$MSE, 10^{-3}$	$T, c$	$MSE, 10^{-3}$	$T, c$	$MSE, 10^{-3}$	$T, c$	$MSE, 10^{-3}$
Единичные	2.55	0.8492	2.76	203.06	2,8	3.7192	2.94	3.8069
Параболические	2.58	0.5479	2,84	201.31	2.83	3,1207	3.07	3.4224
Тригонометрические	2.62	0.7038	2,93	200.69	2.84	3.1259	3,09	3.0222

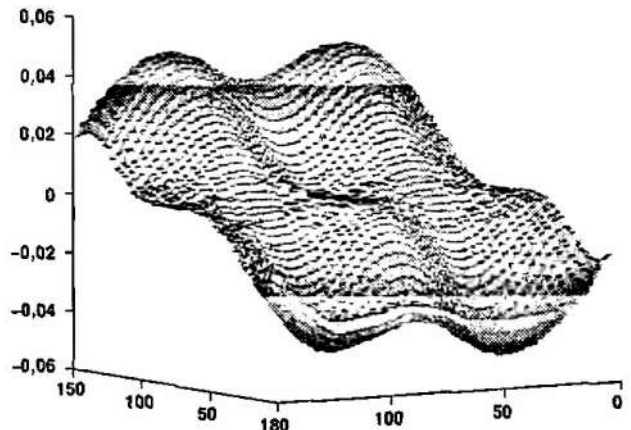


Рис. 6. Аппроксимация сетью СМАС с единичными БФ и нормализацией по частным переменным

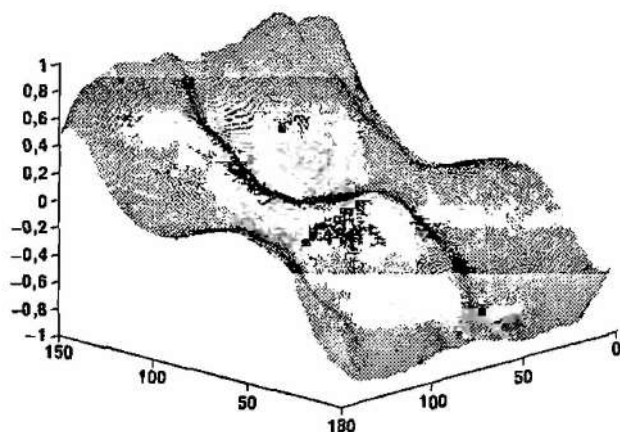


Рис. 7. Аппроксимация сетью СМАС с тригонометрическими БФ с нормализацией по частным переменным и по гиперкубам

НЧП — использовалась нормализация по частным входным переменным. НГК — нормализация по вычисленным значениям гиперкубов и НЧП+НГК — использование обоих видов нормализации.

Очевидно, что нормализация позволяет получать более сглаженные результаты, но при этом значительно ухудшает обучаемость сети. Принципиальной разницы в используемых БФ нет, по крайней мере, в случае применения подобной вычислительной базы. Разница во времени вычислений не столь значительна, так как операции с матрицами создают большую вычислительную нагрузку, чем расчет значений БФ. Тем не менее, разница есть как в значениях ошибки аппроксимации, так и во времени вычислений. Следовательно, БФ параболической формы являются более предпочтительными, так как обеспечивают наилучшие аппроксимирующие свойства сети СМАС при сравнительно небольших вычислительных затратах.

**Эксперимент 2.** Целью следующего эксперимента было исследование аппроксимации многомерных функций сетями СМАС различной архитектуры.

В качестве моделируемой использовалась также функция  $G(x_1, x_2) = F(x_1, 0, 25, 0, 25, x_2)$  и ее окрестности:  $F(x_1, 0, 2 \dots 0, 3, 0, 2 \dots 0, 3, x_2)$  и  $F(x_1, 0, \dots 0, 5, 0, \dots 0, 5, x_2)$ .

Результаты исследований приведены в табл. 2 и на рис. 8–10. Здесь приняты следующие обозначения: НСМАС — иерархическая сеть СМАС, состоящая из трех СМАС (одной — в выходном и двух — в скрытом слоях); LDB6 — LDB СМАС с полным набором парных комбинаций входных переменных, состоящая из 6 сетей СМАС со входами  $(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4),$

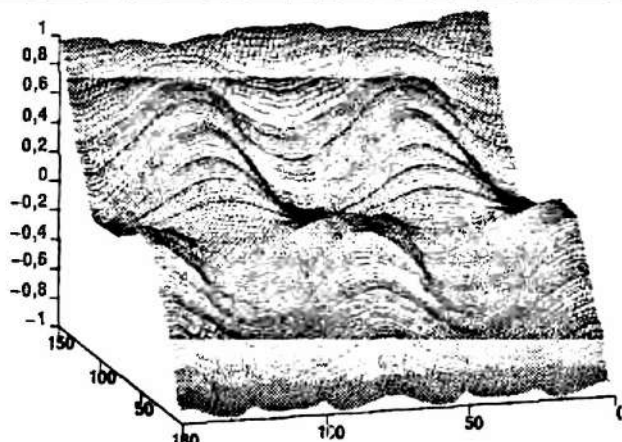


Рис. 8. Аппроксимация функции  $F(x_1, 0, 2 \dots 0, 3, 0, 2 \dots 0, 3, x_2)$  сетью НСМАС с параболическими БФ

$(x_3, x_4)$ ; LDB3 — LDB СМАС, состоящая из трех сетей СМАС со входами  $(x_1, x_2), (x_1, x_4), (x_3, x_4)$ .

Для аппроксимации такой функции обычной четырехмерной СМАС согласно формуле (5) потребовалось бы 12 119 ячеек памяти. Как видно из табл. 2, данные методы позволяют сократить объем требуемой СМАС памяти для хранения весовых коэффициентов примерно в 5–11 раз, сохраняя при этом

Таблица 2

Тип сети	Вид базисных функций	Дисперсия траекторий	Объем памяти, ячеек	T, с	MSE, $10^{-3}$
НСМАС	тригонометрический	0	1050	9,88	0,3534
НСМАС	параболический	0	1050	9,59	0,4176
LDB3	единичный	0	1053	10,21	0,4865
LDB3	тригонометрический	0	1053	10,62	0,1970
LDB3	параболический	0	1053	10,29	0,1528
LDB6	единичный	0	2106	20,21	0,5608
LDB6	тригонометрический	0	2106	20,31	0,1893
LDB6	параболический	0	2106	19,97	0,1274
НСМАС	тригонометрический	0,1	1050	9,79	0,7429
НСМАС	параболический	0,1	1050	9,56	0,7301
LDB3	единичный	0,1	1053	10,14	0,8628
LDB3	тригонометрический	0,1	1053	10,51	0,4083
LDB3	параболический	0,1	1053	10,33	0,2667
LDB6	единичный	0,1	2106	20,05	0,6004
LDB6	тригонометрический	0,1	2106	20,76	0,1751
LDB6	параболический	0,1	2106	20,45	0,1552
НСМАС	тригонометрический	0,5	1050	9,95	3,9341
НСМАС	параболический	0,5	1050	9,55	3,0587
LDB3	единичный	0,5	1053	10,20	2,4866
LDB3	тригонометрический	0,5	1053	10,87	5,6368
LDB3	параболический	0,5	1053	10,32	4,5994
LDB6	единичный	0,5	2106	19,75	1,7994
LDB6	тригонометрический	0,5	2106	20,41	2,3533
LDB6	параболический	0,5	2106	20,05	2,2995

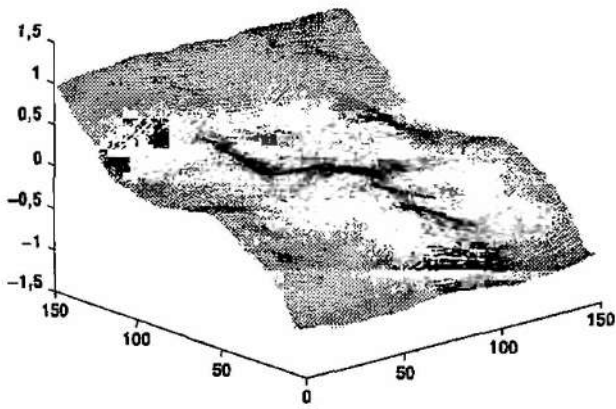


Рис. 9. Аппроксимация функции  $F(x_1, 0...0,5, 0...0,5, x_4)$  сетью LDB3 с тригонометрическими БФ

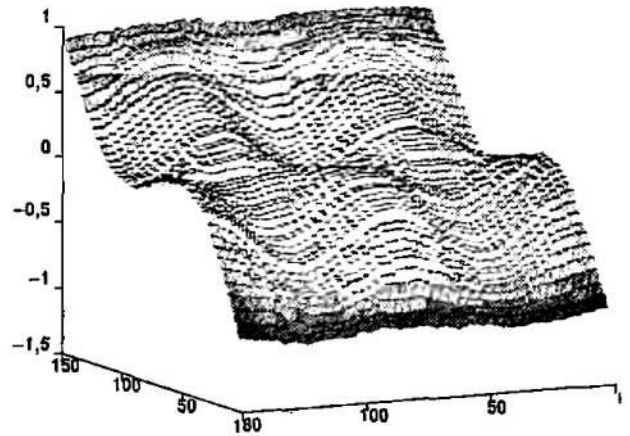


Рис. 10. Аппроксимация функции  $F(x_1, 0...0,5, 0...0,5, x_4)$  сетью LDB6 с тригонометрическими БФ

способность СМАС весьма эффективно аппроксимировать нелинейные многомерные функции.

### 6. Выводы

Результаты исследований свидетельствуют о том, что аппроксимация многомерных функций может быть достаточно эффективно осуществлена путем применения либо перархической (НСМАС), либо линейной (LDB) СМАС, образованной сетями малой размерности, архитектур сетей СМАС.

Если в первом случае используемые БФ должны быть дифференцируемыми, то во втором — произвольными. Среди дифференцируемых наиболее простой и в то же время весьма эффективной является параболическая функция. Использование в данных сетях операции нормализации, приводя к более гладким поверхностям, получаемым в результате аппроксимации, ухудшает обучаемость сети. Однако во всех случаях удается существенно сократить объем требуемой памяти сети.

Список литературы: 1. *Albus J. S.* A new approach to manipulator control: the cerebellar model articulation controller (CMAC) // *ASME Trans., J. Dynamic Systems, Measurement and Control.* — 1975. — № 3. — P. 220–227. 2. *Jou C. -C.* A fuzzy cerebellar model articulation controller // *IEEE Int. Conf. on Fuzzy Systems.* — 1992. — P. 1171–1178. 3. *Руденко О. Г., Бессонов А. А.* О выборе базисных функций в нейронной сети СМАС // *Проблемы управления и информатики.* — 2004. — № 2. — С. 143–155. 4. *Lee H. -M., Chen Ch. -M., Lu Yu. -F.* A Self-organizing HCMAC Neural Network Classifier // *IEEE Transactions on Neural Networks.* — 2003. — 14, № 1. — P. 15–26. 5. *Chiang Ch. -T., Lin Ch. -Sh.* CMAC with General Basis Functions // *Neural Networks.* — 1996. — 9, № 7. — P. 1199–1211. 6. *Lane S. H., Handelman D. A., Gelfand J. J.* Theory and development of higher-order CMAC neural networks // *IEEE Control Systems.* — 1992. — 12, № 2. — P. 23–30. 7. *Wang Z. -Q., Jeffrey L. Schiano, Ginsberg M.* Hash-Coding in CMAC Neural Networks // *IEEE Int. Conf. on Neural Networks.* — 1996. — 3. — P. 1698–1703. 8. *Lin Ch. -Sh., Li Ch. -K.* A Low-Dimensional-CMAC-Based Neural Network // *IEEE Int. Conf. on Neural Networks.* — 1996. — P. 1297–1302.

Поступила в редакцию 28.11.2006