

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

Адаптация системы нечёткого вывода типа Мамдани
по дисциплине «Нечёткие системы обработки информации
в мехатронике и робототехнике»

Выполнил
студент гр. 3341506/10401

Паньков И.С.

Проверил
ассистент

Абросимов Э.А.

«__» _____ 2022 г.

Санкт-Петербург
2022

Цель работы

Цель работы — изучить особенности построения адаптивных нечётких систем с использованием нелинейной оптимизации пакета Optimization Toolbox в среде MATLAB.

Задание

1 Ознакомиться с особенностями использования функции `fmincon` пакета Optimization Toolbox по соответствующему справочному материалу. Данная функция позволяет находить минимум скалярной функции нескольких аргументов при заданном начальном приближении и при наличии линейных и нелинейных ограничений (задача нелинейного программирования).

2 Построить изображения поверхности нелинейной зависимости

$$y = \sin^2(x_1 - 2x_2) \cdot e^{-|x_2|}, \quad x_1 \in [0; \pi], \quad x_2 \in [-1; 1]. \quad (1)$$

и поверхности «входы – выход» аппроксимирующей системы нечёткого (аналогичной той, которая была разработана в лабораторной работе №1).

3 Сформировать обучающий и контрольный массивы данных, которые будут использоваться в процессе оптимизации системы нечёткого вывода. Элементами данных массивов являются точки, равномерно распределённые по области определения исходной нелинейной зависимости (1). Построить графическое изображение распределения точек обучающего и контрольных массивов на изображении поверхности исходной зависимости.

4 Выбрать настраиваемые в процессе оптимизации параметры системы нечёткого вывода. Задать начальные приближения параметров, а также верхние и нижние границы их изменения. Для повышения эффективности работы алгоритма оптимизации ввести масштабирование настраиваемых параметров.

5 Задать основные параметры оптимизации.

6 Разработать функцию обновления параметров системы нечёткого вывода для её модификации в процессе оптимизации.

7 Разработать функцию для определения среднеквадратичной ошибки аппроксимации.

8 Запустить процесс оптимизации системы нечёткого вывода.

9 Построить изображение поверхности «входы – выход» аппроксимирующей системы нечёткого вывода после оптимизации.

10 Определить значения среднеквадратичных ошибок аппроксимации заданной нелинейной зависимости (1) при помощи исходной системы нечёткого вывода и системы нечёткого вывода, полученной после выполнения параметрической оптимизации.

11 Повторить процесс оптимизации системы нечёткого вывода без использования масштабирования параметров и сравнить полученные результаты.

12 Сделать выводы из проделанной работы, в которых отразить:

- область возможного применения оптимизации систем нечёткого вывода для решения прикладных задач;
- особенности выбора оптимизируемых параметров системы нечёткого вывода;
- особенности использования масштабирования настраиваемых параметров в процессе оптимизации.

13 Подготовить отчёт по лабораторной работе.

Ход работы

Исходная нелинейная зависимость (1) описывает поверхность, график которой представлен на рисунке 1.

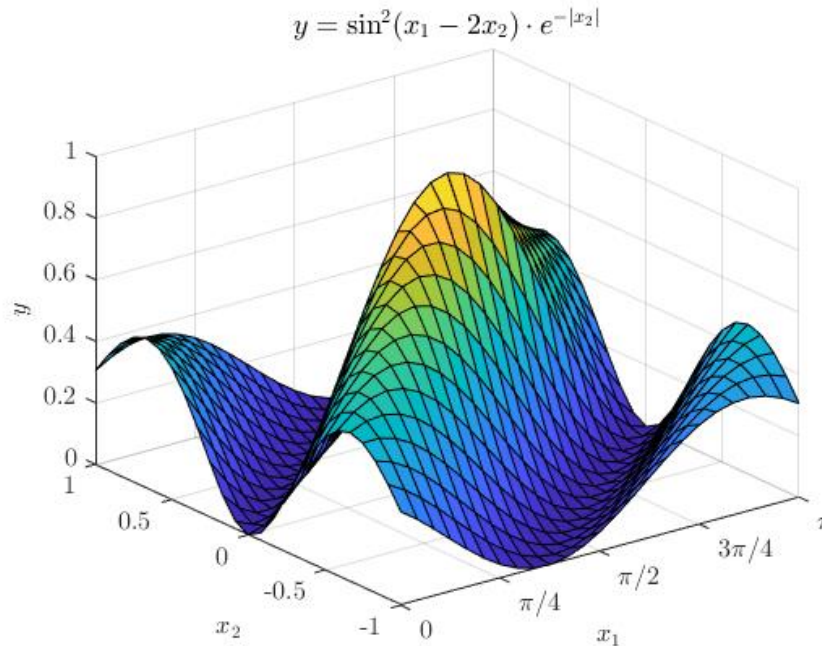


Рисунок 1 — Поверхность исходной нелинейной зависимости

Для аппроксимации зависимости ранее была разработана система нечёткого вывода типа Мамдани. Поскольку поверхность достаточно сложна для описания, было решено создавать систему сразу с пятью термами как входных, так и выходных переменных: negative-big, negative-middle, zero, positive-middle и positive-big. В результате исследования зависимости точности аппроксимации от типа функции принадлежности было решено использовать гауссовы функции принадлежности. Графики функций принадлежности термов входных и выходных переменных $\mu(x_1)$, $\mu(x_2)$ и $\mu(y)$ представлен на рисунке 2.

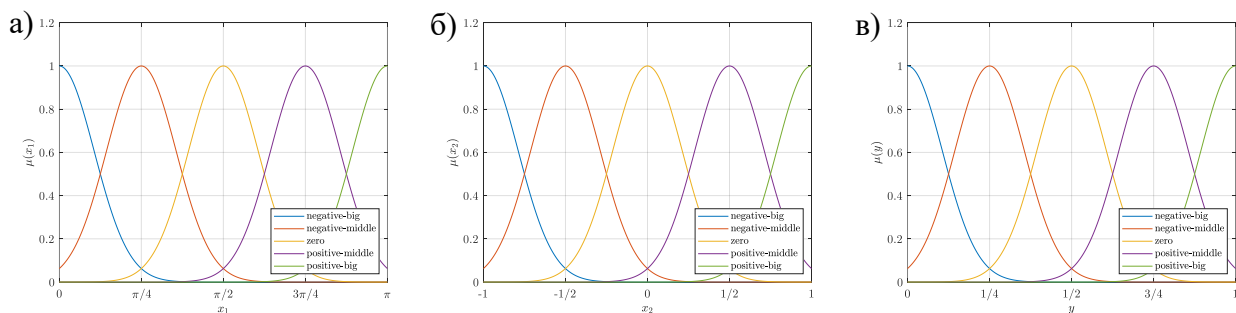


Рисунок 2 — Функции принадлежности термов входных и выходных переменных: а) — $\mu(x_1)$, б) — $\mu(x_2)$, в) — $\mu(y)$

Для системы была определена следующая база правил нечётких продукций:

- 1 Если $x_1 = \text{zero}$ и $x_2 = \text{zero}$, то $y = \text{positive-big}$;
- 2 Если $x_1 = \text{negative-middle}$ и $x_2 = \text{negative-middle}$, то $y = \text{zero}$;
- 3 Если $x_1 = \text{positive-middle}$ и $x_2 = \text{positive-middle}$, то $y = \text{zero}$;
- 4 Если $x_1 = \text{negative-big}$ и $x_2 = \text{positive-big}$, то $y = \text{zero}$;
- 5 Если $x_1 = \text{positive-big}$ и $x_2 = \text{negative-big}$, то $y = \text{zero}$;
- 6 Если $x_1 = \text{negative-big}$ и $x_2 = \text{negative-big}$, то $y = \text{negative-middle}$;
- 7 Если $x_1 = \text{positive-big}$ и $x_2 = \text{positive-big}$, то $y = \text{negative-middle}$;
- 8 Если $x_1 = \text{negative-middle}$ и $x_2 = \text{positive-middle}$, то $y = \text{negative-big}$;
- 9 Если $x_1 = \text{positive-middle}$ и $x_2 = \text{negative-middle}$, то $y = \text{negative-big}$;
- 10 Если $x_1 = \text{zero}$ и $x_2 = \text{negative-big}$, то $y = \text{negative-big}$;
- 11 Если $x_1 = \text{zero}$ и $x_2 = \text{positive-big}$, то $y = \text{negative-big}$;
- 12 Если $x_1 = \text{negative-big}$ и $x_2 = \text{zero}$, то $y = \text{negative-big}$;
- 13 Если $x_1 = \text{positive-big}$ и $x_2 = \text{zero}$, то $y = \text{negative-big}$.

В итоге была получена система нечёткого вывода для аппроксимации исходной зависимости (1) со среднеквадратичной ошибкой $\text{RMSE} = 0,1396$, поверхность «входы – выход» которой представлена на рисунке 3.

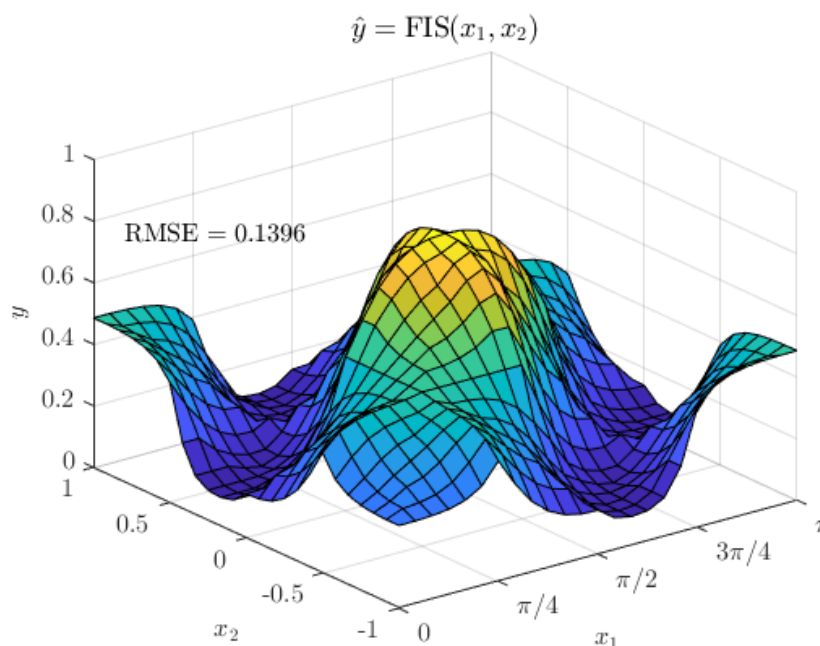


Рисунок 3 — Поверхность «входы – выход» системы нечёткого вывода

Оптимизируем полученную аппроксимирующую систему нечёткого вывода по критерию минимизации среднеквадратичной ошибки аппроксимации. В первую очередь сформируем обучающий и контрольный массивы данных, состоящие из точек, принадлежащих исходной поверхности и равномерно распределённых по области её определения. Графики распределения точек обучающей и контрольной выборок представлены на рисунке 4.

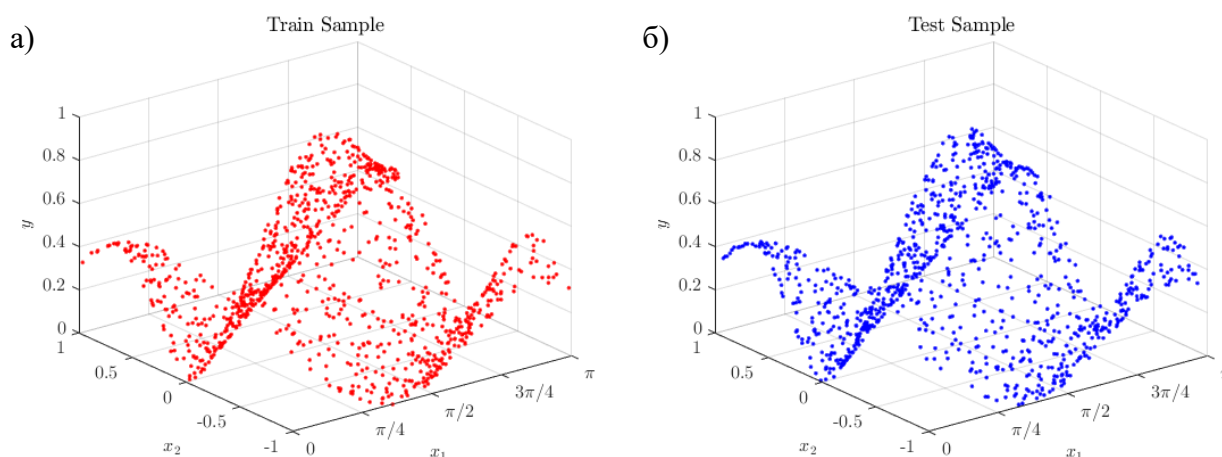


Рисунок 4 — Распределение точек: а) — обучающей выборки, б) — контрольной выборки

Выберем параметры системы для оптимизации. Ранее было замечено, что система обладает осевой симметрией относительно вертикальной прямой, проходящей через точку $(0; \pi/2)$ на горизонтальной плоскости. Используем этот факт, для сокращения количества настраиваемых параметров: очевидно, что параметры симметричных функций принадлежности, например функций принадлежности термов входных переменных *negative-middle* и *positive-middle*, будут одинаковыми.

Исходя из вышесказанного в качестве параметров для оптимизации функций входных переменных выберем дисперсии функций принадлежности термов *negative-big*, *negative-middle* и *zero*, а также математическое ожидание терма *positive-middle*, то есть восемь параметров — по четыре для каждой переменной.

В отличие от входных переменных, выходная переменная не обладает какой-либо выраженной симметрией, однако можно обратить внимание, что функция принадлежности терма *positive-middle* не использована ни в одном продукционном правиле, а значит, её параметры могут быть проигнорированы.

Таким образом, в качестве параметров для оптимизации функций выходных переменных выберем дисперсии функций принадлежности термов *negative-big*, *negative-middle*, *zero* и *positive-big*, а также математические ожидания термов *negative-middle* и *zero*, то есть ещё шесть параметров.

База правил нечётких продукций системы нечёткого вывода также обладает симметрией: все правила, кроме правила 1, сгруппированы по два и симметричны относительно входных переменных. Таким образом, оба сгруппированных правила должны обладать одинаковыми весовыми коэффициентами.

В качестве параметров для оптимизации базы правил нечётких продукций выберем весовой коэффициент правила 1, а также весовые коэффициенты пар правил (2, 3), (4, 5) и так далее, то есть ещё семь параметров. Как итог имеем 21 параметр для оптимизации.

В качестве ограничений параметров для оптимизации выберем отклонение $\pm 30\%$ для параметров функций принадлежности. Ограничения для весовых коэффициентов продукционных правил очевидны: их значения должны лежать в диапазоне $[0; 1]$. Все ранее упомянутые параметры имеют один порядок, но для целей масштабирования можно домножить параметры первой и второй входных переменных на $1/\pi$ и $1/2$ соответственно. В дальнейшем будет использовано два способа оптимизации: с масштабированием параметров и без него.

Разработаем функцию для обновления параметров системы нечёткого вывода `update_fis_params`. Данная функция принимает на вход три параметра: систему нечёткого вывода `fis`, вектор оптимизационных параметров системы `params` и вектор коэффициентов масштабирования `scale`, причём последний параметр является опциональным и по умолчанию равен вектору из единиц той же длины, что и вектор `params`. Функция модифицирует значения дисперсии и математического ожидания функций принадлежности термов входных и выходных переменных и весовые коэффициенты продукционных правил исходной системы. После этого функция возвращает обновлённую систему нечёткого вывода. Реализация функции языке MATLAB представлена в листинге 1.

Листинг 1 — Функция update_fis_params

```

001 function fis = update_fis_params(fis, params, scale)
002     if (nargin < 3 || isempty(scale) == true)
003         scale = ones(1, length(params));
004     end
005     params = params ./ scale;
006
007     fis.input(1).mf(1).params(1) = params(1);
008     fis.input(1).mf(2).params(1) = params(2);
009     fis.input(1).mf(3).params(1) = params(3);
010     fis.input(1).mf(4).params(1) = params(2);
011     fis.input(1).mf(5).params(1) = params(1);
012     fis.input(1).mf(2).params(2) = pi / 2 - params(4);
013     fis.input(1).mf(4).params(2) = pi / 2 + params(4);
014
015     fis.input(2).mf(1).params(1) = params(5);
016     fis.input(2).mf(2).params(1) = params(6);
017     fis.input(2).mf(3).params(1) = params(7);
018     fis.input(2).mf(4).params(1) = params(6);
019     fis.input(2).mf(5).params(1) = params(5);
020     fis.input(2).mf(2).params(2) = 0 - params(8);
021     fis.input(2).mf(4).params(2) = 0 + params(8);
022
023     fis.output.mf(1).params(1) = params(9);
024     fis.output.mf(2).params(1) = params(10);
025     fis.output.mf(3).params(1) = params(11);
026     fis.output.mf(5).params(1) = params(12);
027     fis.output.mf(2).params(2) = params(13);
028     fis.output.mf(3).params(2) = params(14);
029
030     fis.rule(1).weight = params(15);
031     fis.rule(2).weight = params(16);
032     fis.rule(3).weight = params(16);
033     fis.rule(4).weight = params(17);
034     fis.rule(5).weight = params(17);
035     fis.rule(6).weight = params(18);
036     fis.rule(7).weight = params(18);
037     fis.rule(8).weight = params(19);
038     fis.rule(9).weight = params(19);
039     fis.rule(10).weight = params(20);
040     fis.rule(11).weight = params(20);
041     fis.rule(12).weight = params(21);
042     fis.rule(13).weight = params(21);
043 end
044

```

Создадим скрипт для задания начальных приближений оптимизационных параметров, верхних и нижних границ их изменения и непосредственно оптимизации системы нечёткого вывода — как с масштабированием параметров, так и без. Функция для оптимизации `rmse` может быть задана в виде анонимной функции на языке MATLAB. Команды для построения изображений поверхностей оформим в виде отдельных функций. Скрипт для загрузки исходной системы нечёткого вывода, её оптимизации по критерию минимизации среднеквадратичной ошибки аппроксимации, а также тестирования и сравнения систем, оптимизированных различными способами представлен в листинге 2.

Листинг 2 — Скрипт для загрузки исходной системы нечёткого вывода, её оптимизации, а также тестирования и сравнения систем нечёткого вывода

```

001 clc; clear; close all;
002
003 n = 25;
004 x1_min = 0;
005 x1_max = pi;
006 x2_min = -1;
007 x2_max = 1;
008 y_min = 0;
009 y_max = 1;
010
011 x1 = linspace(x1_min, x1_max, n);
012 x2 = linspace(x2_min, x2_max, n);
013 x = reshape(cat(3, repmat(x1, length(x2), 1)', ...
014               repmat(x2, length(x1), 1)), [], 2, 1);
015
016 f = @(x1, x2) sin(x1 - 2 * x2).^2 .* exp(-abs(x2));
017 y = f(x1, x2');
018 print_surface_plot(x1, x2, y, 'Original Function', 'original_function.emf');
019 fis1 = readfis('..\model\mamdani_gaussmf_5in_gaussmf_5out.fis');
020
021 n_train = 1000;
022 n_test = 1000;
023
024 x1_train = x1_min + (x1_max - x1_min) * rand(n_train, 1)';
025 x1_test = x1_min + (x1_max - x1_min) * rand(n_test, 1)';
026
027 x2_train = x2_min + (x2_max - x2_min) * rand(n_train, 1)';
028 x2_test = x2_min + (x2_max - x2_min) * rand(n_test, 1)';
029
030 x_train = [x1_train; x2_train];
031 x_test = [x1_test; x2_test];
032
033 y_train = f(x_train(1, :), x_train(2, :))';
034 y_test = f(x_test(1, :), x_test(2, :))';
035
036 x1_disp = [fis1.inputs(1).mf(1).params(1);
037            fis1.inputs(1).mf(2).params(1);
038            fis1.inputs(1).mf(3).params(1)];
039
040 x1_mean = fis1.inputs(1).mf(4).params(2) - mean([x1_min, x1_max]);
041
042 x2_disp = [fis1.inputs(2).mf(1).params(1);
043            fis1.inputs(2).mf(2).params(1);
044            fis1.inputs(2).mf(3).params(1)];
045
046 x2_mean = fis1.inputs(2).mf(4).params(2) - mean([x2_min, x2_max]);
047
048 y_disp = [fis1.outputs.mf(1).params(1);
049           fis1.outputs.mf(2).params(1);
050           fis1.outputs.mf(3).params(1);
051           fis1.outputs.mf(5).params(1)];
052 y_mean = [fis1.outputs.mf(2).params(2);
053           fis1.outputs.mf(3).params(2)];
054
055 x1_disp_lower = 0.3 * x1_disp;
056 x2_disp_lower = 0.3 * x2_disp;
057 y_disp_lower = 0.3 * y_disp;
058
059 x1_disp_upper = 1.3 * x1_disp;
060 x2_disp_upper = 1.3 * x2_disp;
061 y_disp_upper = 1.3 * y_disp;
062

```

Продолжение листинга 2

```

063 x1_mean_lower = x1_mean - 0.3 * (x1_max - x1_min);
064 x2_mean_lower = x2_mean - 0.3 * (x2_max - x2_min);
065 y_mean_lower = y_mean - 0.3 * (y_max - y_min);
066
067 x1_mean_upper = x1_mean + 0.3 * (x1_max - x1_min);
068 x2_mean_upper = x2_mean + 0.3 * (x2_max - x2_min);
069 y_mean_upper = y_mean + 0.3 * (y_max - y_min);
070
071 w_rule = 0.95 * ones(7, 1);
072 w_rule_lower = zeros(7, 1);
073 w_rule_upper = ones(7, 1);
074
075 params0 = [x1_disp; x1_mean;
076            x2_disp; x2_mean;
077            y_disp; y_mean; w_rule]';
078
079 lower = [x1_disp_lower; x1_mean_lower;
080          x2_disp_lower; x2_mean_lower;
091          y_disp_lower; y_mean_lower; w_rule_lower]';
092
093 upper = [x1_disp_upper; x1_mean_upper;
094          x2_disp_upper; x2_mean_upper;
095          y_disp_upper; y_mean_upper; w_rule_upper]';
096
097 rmse = @(p, s, fis, x, y) ...
098       sqrt(sum(sum((y - evalfis(update_fis_params(fis, p, s), x)).^2)) / numel(y));
099 options = optimset('Display', 'iter', 'MaxIter', 25);
100
101 scale = [1/pi, 1/pi, 1/pi, 1/pi, 1/2, 1/2, 1/2, 1/2, ones(1, 13)];
102 params = fmincon(rmse, scale .* params0, [], [], [], [], ...
103                scale .* lower, scale .* upper, [], ...
104                options, scale, fis1, x_train, y_train);
105 fis2 = update_fis_params(fis1, params, scale);
106
107 scale = ones(1, 21);
108 params = fmincon(rmse, scale .* params0, [], [], [], [], ...
109                scale .* lower, scale .* upper, [], ...
110                options, scale, fis1, x_train, y_train);
111 fis3 = update_fis_params(fis1, params, scale);
112
113 y1 = reshape(evalfis(fis1, x), length(x1), length(x2))';
114 rmse1 = sqrt(sum(sum((y_test - evalfis(fis1, x_test)).^2)) / numel(y_test));
115 print_surface_plot(x1, x2, y1, 'Mamdani FIS Before Optimization', ...
116                    'mamdani_gauss_5in_gauss_5out_surface_default.emf', rmse1);
117
118 y2 = reshape(evalfis(fis2, x), length(x1), length(x2))';
119 rmse2 = sqrt(sum(sum((y_test - evalfis(fis2, x_test)).^2)) / numel(y_test));
120 print_surface_plot(x1, x2, y2, 'Mamdani FIS After Optimization With Scale', ...
121                    'mamdani_gauss_5in_gauss_5out_surface_custom.emf', rmse2);
122
123 y3 = reshape(evalfis(fis3, x), length(x1), length(x2))';
124 rmse3 = sqrt(sum(sum((y_test - evalfis(fis3, x_test)).^2)) / numel(y_test));
125 print_surface_plot(x1, x2, y3, 'Mamdani FIS After Optimization Without Scale', ...
126                    'mamdani_gauss_5in_gauss_5out_surface_ custom_no_scale.emf',
127                    rmse3);
128
129
130
131
132
133
134
135

```

Поверхности «входы – выход» систем нечёткого вывода, полученных в результате оптимизации с использованием и без использования масштабирования параметров представлены на рисунке 5.

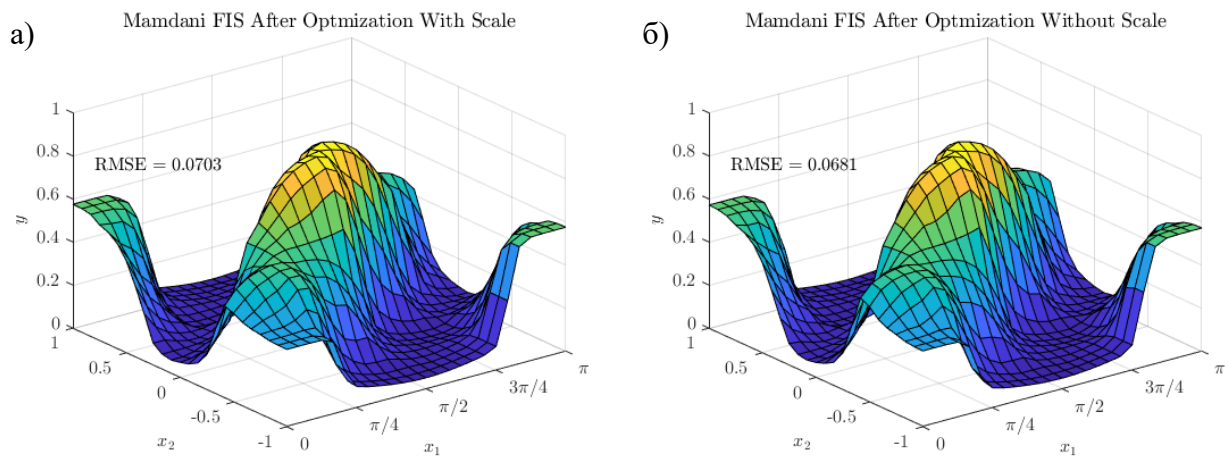


Рисунок 5 — Поверхности «входы – выход» систем нечёткого вывода:

- а) — после оптимизации с использованием масштабирования параметров,
- б) — после оптимизации без использования масштабирования параметров

Как видно по значениям среднеквадратичной ошибки ($RMSE = 0,0703$ и $RMSE = 0,0681$), полученные в результате оптимизации аппроксимирующие системы нечёткого вывода позволяют достигнуть более чем в два раза лучшей точности аппроксимации исходной зависимости. При этом можно обратить внимание, что в данном случае масштабирование параметров оптимизации не даёт положительного эффекта и даже напротив несколько снижает точность итоговой системы при одинаковом числе итераций оптимизационного процесса.

Вывод

Оптимизация, или адаптация, системы нечёткого вывода под конкретную задачу позволяет кратно увеличить точность работы системы без необходимости её кардинально перерабатывать (например, менять типы функций принадлежности или продукционные правила). Однако в этом случае важным аспектом качества оптимизации является выбор параметров для оптимизации. При грамотном подборе этих параметров в процессе анализа задачи может не только увеличиться скорость оптимизации, но и сохранена «прозрачность» работы нечёткой системы. Также необходимо осторожно подходить к масштабированию оптимизационных параметров, поскольку в некоторых случаях оно может не только не дать ощутимого положительного результата, но и снизить итоговую точность оптимизируемой системы.

23:46

24 ноября 2022