МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Нечеткая система Мамдани

Вариант 8

Отчёт по лабораторной работе №8 дисциплины

«Системы обработки знаний»

Выполнил студент группы ИВТ-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Жеребцов К. А.   
Проверил доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Ростовцев В. С.

Киров 2024

1. Задание

Создать систему нечеткого вывода Мамдани, которая моделирует зависимость y = y = 3\*x13 \*cos(x2-4)

Диапазон x1 = [-6; 5]

Диапазон x2 = [-6;4]

Построить график функции, выполнить моделирование со всеми типами функций принадлежности и выбрать ее оптимальный тип по критерию минимума среднеквадратичного отклонения.

Для алгоритма Мамдани выполнить моделирование со следующими параметрами:

* метод агрегации (максимум, сумма, вероятностное «ИЛИ»);
* метод дефаззификации (центр тяжести, медиана, наибольший из максимумов);
* метод «И» (минимум, умножение (вероятностное «И»));
* метод «ИЛИ» (максимум, вероятностное «ИЛИ»);
* метод импликации (минимум, умножение);

Выбрать оптимальные параметры нечеткой системы по критерию минимума среднеквадратичного отклонения.

По результатам проектирования системы нечёткого вывода в Fuzzy Logic Toolbox составить отчет в электронном виде, включив в него результаты промежуточных этапов, а также графики моделируемой и аппроксимированной функции, сформировать выводы.

1. Выполнение лабораторной работы

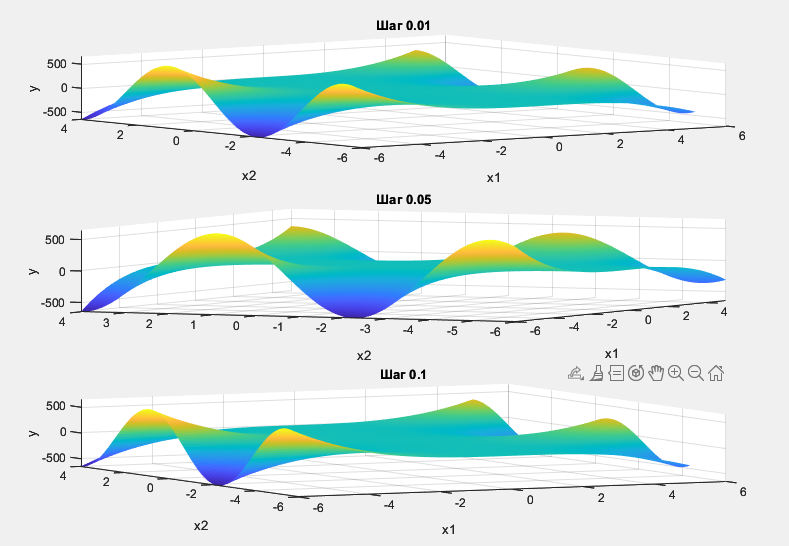


Рисунок 1 – Графики функции, полученные в ходе лабораторной работы №7

Нечеткие правила, полученные в ходе лабораторной работы №7:

Если x1 = «Низкий» и x2 = «Низкий», то y = «Высокий»

Если x1 = «Низкий» и x2 = «Средний», то y = «Низкий»

Если x1 = «Низкий» и x2 = «Высокий», то y = «Низкий»

Если x1 = «Средний» и x2 = «Низкий», то y = «Средний»

Если x1 = «Средний» и x2 = «Средний», то y = «Средний»

Если x1 = «Средний» и x2 = «Высокий», то y = «Средний»

Если x1 = «Высокий» и x2 = «Низкий», то y = «Низкий»

Если x1 = «Высокий» и x2 = «Средний», то y = «Низкий»

Если x1 = «Высокий» и x2 = «Высокий», то y = «Высокий»

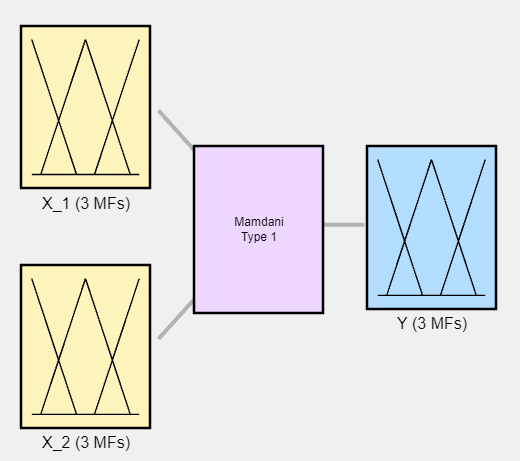


Рисунок 2 – Система Мамдани

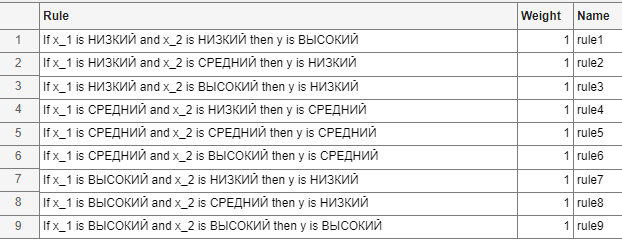


Рисунок 3 – Правила

Исходный код для вычисления среднеквадратичной ошибки

>> f = @(x1, x2) 3 \* x1.^3 .\* cos(x2 - 4);

types = ["gaussmf", "gauss2mf", "trimf", "trapmf", "gbellmf"];

types\_errors = {};

for i = 1:length(types)

for j = 1:length(mamdani.inputs)

for k = 1:length(mamdani.inputs(j).mf)

mamdani.inputs(j).mf(k).type = types(i);

end

end

for j = 1:length(mamdani.outputs)

for k = 1:length(mamdani.outputs(j).mf)

mamdani.outputs(j).mf(k).type = types(i);

end

end

%fuzzyLogicDesigner(mamdani);

%pause;

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp(types(i) + " " + E);

types\_errors{end+1} = [types(i), E];

end

best\_type = types\_errors{1};

for i = 2:length(types\_errors)

if types\_errors{i}(2) < best\_type(2)

best\_type = types\_errors{i};

end

end

disp("Best type: " + best\_type(1) + " " + best\_type(2));

for j = 1:length(mamdani.inputs)

for k = 1:length(mamdani.inputs(j).mf)

mamdani.inputs(j).mf(k).type = best\_type(1);

end

end

for j = 1:length(mamdani.outputs)

for k = 1:length(mamdani.outputs(j).mf)

mamdani.outputs(j).mf(k).type = best\_type(1);

end

end

fuzzyLogicDesigner(mamdani);

Таблица 1 – результаты изменения типа входов и выхода

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| gaussmf | gauss2mf | trimf | trapmf | gbellmf |
| 31086.6708 | 30640.1389 | 31106.4037 | 30805.0463 | 30579.1738 |

Наибольшей точностью из вышерассмотренных обладает функция gbellmf

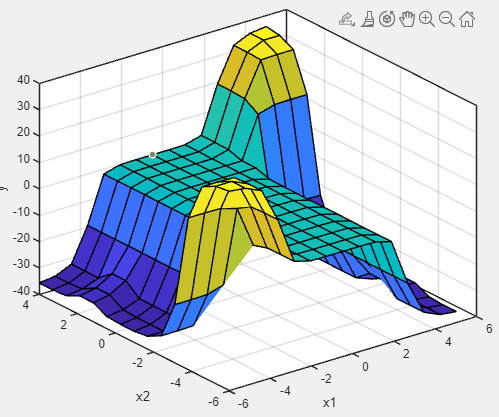


Рисунок 5 – График функции с применением gbellmf для входов и выходов

Исходный код поэтапного подбора параметров:

>> best\_type = "gbellmf";

for i = 1:length(mamdani.inputs)

for j = 1:length(mamdani.inputs(i).mf)

mamdani.inputs(i).mf(j).type = best\_type;

end

end

for i = 1:length(mamdani.outputs)

for j = 1:length(mamdani.outputs(i).mf)

mamdani.outputs(i).mf(j).type = best\_type;

end

end

%fuzzyLogicDesigner(mamdani);

%pause;

and\_methods = ["min", "prod"];

or\_methods = ["max", "probor"];

implication\_methods = ["min", "prod"];

aggregation\_methods = ["max", "probor"];

defuzz\_methods = ["centroid", "bisector", "lom", "mom", "som"];

best\_and\_method = "";

best\_and\_error = intmax;

for i = 1:length(and\_methods)

mamdani.AndMethod = and\_methods(i);

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp(and\_methods(i) + " error: " + E);

if E < best\_and\_error

best\_and\_error = E;

best\_and\_method = and\_methods(i);

end

end

mamdani.AndMethod = best\_and\_method;

best\_or\_method = "";

best\_or\_error = intmax;

for i = 1:length(or\_methods)

mamdani.OrMethod = or\_methods(i);

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp(or\_methods(i) + " error: " + E);

if E < best\_or\_error

best\_or\_error = E;

best\_or\_method = or\_methods(i);

end

end

mamdani.OrMethod = best\_or\_method;

best\_implication\_method = "";

best\_implication\_error = intmax;

for i = 1:length(implication\_methods)

mamdani.ImplicationMethod = implication\_methods(i);

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp(implication\_methods(i) + " error: " + E);

if E < best\_implication\_error

best\_implication\_error = E;

best\_implication\_method = implication\_methods(i);

end

end

mamdani.ImplicationMethod = best\_implication\_method;

best\_aggregation\_method = "";

best\_aggregation\_error = intmax;

for i = 1:length(aggregation\_methods)

mamdani.AggregationMethod = aggregation\_methods(i);

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp(aggregation\_methods(i) + " error: " + E);

if E < best\_aggregation\_error

best\_aggregation\_error = E;

best\_aggregation\_method = aggregation\_methods(i);

end

end

mamdani.AggregationMethod = best\_aggregation\_method;

best\_defuzz\_method = "";

best\_defuzz\_error = intmax;

for i = 1:length(defuzz\_methods)

mamdani.DefuzzMethod = defuzz\_methods(i);

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp(defuzz\_methods(i) + " error: " + E);

if E < best\_defuzz\_error

best\_defuzz\_error = E;

best\_defuzz\_method = defuzz\_methods(i);

end

end

mamdani.DefuzzMethod = best\_defuzz\_method;

%fuzzyLogicDesigner(mamdani);

%pause;

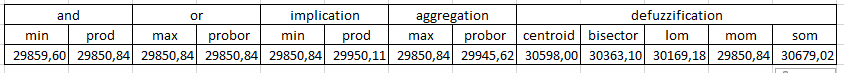
[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp("Final error: " + E);

Таблица 2 – Результат подбора



Подбор выполнялся поэтапно, после каждого этапа сохранялся лучший результат предыдущего параметра.

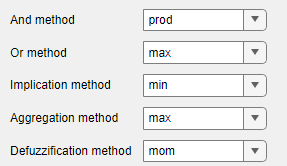


Рисунок 9 – Лучшие параметры методом поэтапного подбора

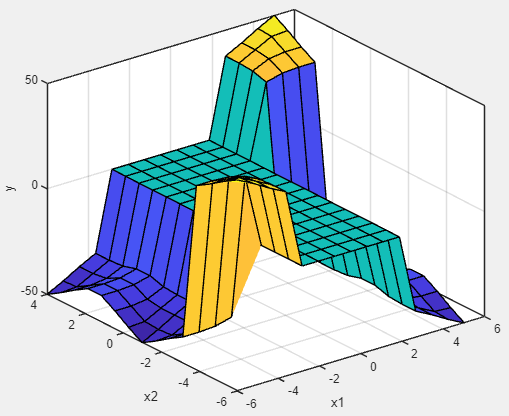


Рисунок 10 – График, полученный в результате поэтапного подбора параметров

Итоговая квадратичная ошибка составила - 29850.8351.

Также был составлен скрипт, реализующий аналог GridSearchCV из библиотеки Python – sklearn.

Данный скрипт делает полный перебор всех возможных вариаций значений параметров и получает лучший результат.

Исходный код метода полного перебора:

for i = 1:length(mamdani.inputs)

for j = 1:length(mamdani.inputs(i).mf)

mamdani.inputs(i).mf(j).type = best\_type;

end

end

for i = 1:length(mamdani.outputs)

for j = 1:length(mamdani.outputs(i).mf)

mamdani.outputs(i).mf(j).type = best\_type;

end

end

fuzzyLogicDesigner(mamdani);

pause;

and\_methods = ["min", "prod"];

or\_methods = ["max", "probor"];

implication\_methods = ["min", "prod"];

aggregation\_methods = ["max", "probor"];

defuzz\_methods = ["centroid", "bisector", "lom", "mom", "som"];

best\_methods = [];

best\_error = intmax;

for and\_method = and\_methods

for or\_method = or\_methods

for implication\_method = implication\_methods

for aggregation\_method = aggregation\_methods

for defuzz\_method = defuzz\_methods

mamdani.AndMethod = and\_method;

mamdani.OrMethod = or\_method;

mamdani.ImplicationMethod = implication\_method;

mamdani.AggregationMethod = aggregation\_method;

mamdani.DefuzzMethod = defuzz\_method;

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

error = immse(z, y);

if error < best\_error

best\_error = error;

best\_methods = [and\_method, or\_method, implication\_method, aggregation\_method, defuzz\_method];

end

end

end

end

end

end

mamdani.AndMethod = best\_methods(1);

mamdani.OrMethod = best\_methods(2);

mamdani.ImplicationMethod = best\_methods(3);

mamdani.AggregationMethod = best\_methods(4);

mamdani.DefuzzMethod = best\_methods(5);

fuzzyLogicDesigner(mamdani);

pause;

[x1, x2, z] = gensurf(mamdani);

y = f(x1, x2);

E = immse(z, y);

disp("Final error: " + E);

Рисунок 12 – 2 часть метода полного перебора

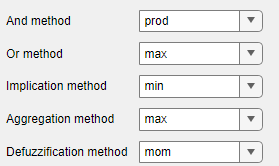


Рисунок 13 – Лучшие параметры методом полного перебора

Итоговая ошибка, полученная методом полного перебора, составила 29850.8351, что не дало улучшения по сравнению с поэтапным подбором параметров.

Выводы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| gaussmf | gauss2mf | trimf | trapmf | gbellmf |
| 31086.6708 | 30640.1389 | 31106.4037 | 30805.0463 | 30579.1738 |

В ходе лабораторной работы была составлена система Мамдани. Опасения, высказанные в выводе лабораторной работы №7, подтвердились. Среднеквадратичная ошибка равна 29850.8351 в лучшем случае, что недостаточно. Из полученного графика видно, что он уступает в точности графику моделируемой функции. Это связано с недостатком количества правил (9), которые были использованы при составлении системы Мамдани.

Были подобраны функции, а также параметры моделирования.

Для подбора параметров было составлено 2 скрипта – для поэтапного подбора и для полного перебора, полный перебор не дал прироста точности

Результат экспериментов показал, что наибольшую эффективность среди используемых функций принадлежности имеет функция gbellmf. Среднеквадратичная ошибка этой функции составила 30579.1738, наихудший результат показала функция trimf со среднеквадратичной ошибкой 31106.4037.

Полный перебор, реализованный скриптом, позволил подобрать лучшие параметры аппроксимации (рис. 13).

После подбора оптимальных параметров среднеквадратичная ошибка составила 29850.8351.