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 Toolbox Optimization 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



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

1. 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 . Построить графическое изображение распределения точек обучающего и контрольных массивов на изображении поверхности исходной зависимости.
2. 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.
3. Задать основные параметры оптимизации.
4. Разработать функцию обновления параметров системы нечёткого вывода для её модификации в процессе оптимизации.
5. Разработать функцию для определения среднеквадратичной ошибки аппроксимации.
6. Запустить процесс оптимизации системы нечёткого вывода.
7. Построить изображение поверхности «входы – выход» аппроксимирующей системы нечёткого вывода после оптимизации.
8. Определить значения среднеквадратичных ошибок аппроксимации заданной нелинейной зависимости при помощи исходной системы нечёткого вывода и системы нечёткого вывода, полученной после выполнения параметрической оптимизации.
9. Повторить процесс оптимизации системы нечёткого вывода без использования масштабирования параметров и сравнить полученные результаты.
10. 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.

1. Подготовить отчёт по лабораторной работе.

# Ход работы

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



Рисунок — 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 ,  и  представлен на рисунке Рисунок 2.

а)

б)

в)



Рисунок — Функции принадлежности термов входных  
и выходных переменных: а) —, б) — , в) — 

Для системы была определена следующая база правил нечётких продукций:

1. Если  zero и  zero, то  positive-big;
2. Если  negative-middle и  negative-middle, то  zero;
3. Если  positive-middle и  positive-middle, то  zero;
4. Если  negative-big и  positive-big, то  zero;
5. Если  positive-big и  negative-big, то  zero;
6. Если  negative-big и  negative-big, то  negative-middle;
7. Если  positive-big и  positive-big, то  negative-middle;
8. Если  negative-middle и  positive-middle, то  negative-big;
9. Если  positive-middle и  negative-middle, то  negative-big;
10. Если  zero и  negative-big, то  negative-big;
11. Если  zero и  positive-big, то  negative-big;
12. Если  negative-big и  zero, то  negative-big;
13. Если  positive-big и  zero, то  negative-big.

В итоге была получена система нечёткого вывода для аппроксимации исходной зависимости со среднеквадратичной ошибкой , поверхность «входы – выход» которой представлена на рисунке Рисунок 3.



Рисунок — 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 Рисунок 4.



а)

б)

Рисунок — 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  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 относительной 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  для параметров функций принадлежности. Ограничения для весовых коэффициентов продукционных правил очевидны: их значения должны лежать в диапазоне . Все ранее упомянутые параметры имеют один порядок, но для целей масштабирования можно домножить параметры первой и второй входных переменных на  и  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 Листинг 1.

Листинг 1 — Функция update\_fis\_params

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

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 Листинг 2.

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

|  |  |
| --- | --- |
| 001  002  003  004  005  006  007  008  009  010  011  012  013  014  015  016  017  018  019  020  021  022  023  024  025  026  027  028  029  030  031  032  033  034  035  036  037  038  039  040  041  042  043  044  045  046  047  048  049  050  051  052  053  054  055  056  057  058  059  060  061  062 | clc; clear; close all;  n = 25;  x1\_min = 0;  x1\_max = pi;  x2\_min = -1;  x2\_max = 1;  y\_min = 0;  y\_max = 1;  x1 = linspace(x1\_min, x1\_max, n);  x2 = linspace(x2\_min, x2\_max, n);  x = reshape(cat(3, repmat(x1, length(x2), 1)', ...  repmat(x2, length(x1), 1)), [], 2, 1);  f = @(x1, x2) sin(x1 - 2 \* x2).^2 .\* exp(-abs(x2));  y = f(x1, x2');  print\_surface\_plot(x1, x2, y, 'Original Function', 'original\_function.emf');  fis1 = readfis('../model/mamdani\_gaussmf\_5in\_gaussmf\_5out.fis');  n\_train = 1000;  n\_test = 1000;  x1\_train = x1\_min + (x1\_max - x1\_min) \* rand(n\_train, 1)';  x1\_test = x1\_min + (x1\_max - x1\_min) \* rand(n\_test, 1)';  x2\_train = x2\_min + (x2\_max - x2\_min) \* rand(n\_train, 1)';  x2\_test = x2\_min + (x2\_max - x2\_min) \* rand(n\_test, 1)';  x\_train = [x1\_train; x2\_train];  x\_test = [x1\_test; x2\_test ];  y\_train = f(x\_train(1, :), x\_train(2, :))';  y\_test = f(x\_test (1, :), x\_test (2, :))';  x1\_disp = [fis1.inputs(1).mf(1).params(1);  fis1.inputs(1).mf(2).params(1);  fis1.inputs(1).mf(3).params(1)];  x1\_mean = fis1.inputs(1).mf(4).params(2) - mean([x1\_min, x1\_max]);  x2\_disp = [fis1.inputs(2).mf(1).params(1);  fis1.inputs(2).mf(2).params(1);  fis1.inputs(2).mf(3).params(1)];  x2\_mean = fis1.inputs(2).mf(4).params(2) - mean([x2\_min, x2\_max]);  y\_disp = [fis1.outputs.mf(1).params(1);  fis1.outputs.mf(2).params(1);  fis1.outputs.mf(3).params(1);  fis1.outputs.mf(5).params(1)];  y\_mean = [fis1.outputs.mf(2).params(2);  fis1.outputs.mf(3).params(2)];  x1\_disp\_lower = 0.3 \* x1\_disp;  x2\_disp\_lower = 0.3 \* x2\_disp;  y\_disp\_lower = 0.3 \* y\_disp;  x1\_disp\_upper = 1.3 \* x1\_disp;  x2\_disp\_upper = 1.3 \* x2\_disp;  y\_disp\_upper = 1.3 \* y\_disp; |

Продолжение листинга Листинг 2

|  |  |
| --- | --- |
| 063  064  065  066  067  068  069  070  071  072  073  074  075  076  077  078  079  080  091  092  093  094  095  096  097  098  099  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135 | x1\_mean\_lower = x1\_mean - 0.3 \* (x1\_max - x1\_min);  x2\_mean\_lower = x2\_mean - 0.3 \* (x2\_max - x2\_min);  y\_mean\_lower = y\_mean - 0.3 \* ( y\_max - y\_min);  x1\_mean\_upper = x1\_mean + 0.3 \* (x1\_max - x1\_min);  x2\_mean\_upper = x2\_mean + 0.3 \* (x2\_max - x2\_min);  y\_mean\_upper = y\_mean + 0.3 \* ( y\_max - y\_min);  w\_rule = 0.95 \* ones(7, 1);  w\_rule\_lower = zeros(7, 1);  w\_rule\_upper = ones(7, 1);  params0 = [x1\_disp; x1\_mean;  x2\_disp; x2\_mean;  y\_disp; y\_mean; w\_rule]';  lower = [x1\_disp\_lower; x1\_mean\_lower;  x2\_disp\_lower; x2\_mean\_lower;  y\_disp\_lower; y\_mean\_lower; w\_rule\_lower]';  upper = [x1\_disp\_upper; x1\_mean\_upper;  x2\_disp\_upper; x2\_mean\_upper;  y\_disp\_upper; y\_mean\_upper; w\_rule\_upper]';  rmse = @(p, s, fis, x, y) ...  sqrt(sum(sum((y - evalfis(update\_fis\_params(fis, p, s), x)).^2)) / numel(y));  options = optimset('Display', 'iter', 'MaxIter', 25);  scale = [1/pi, 1/pi, 1/pi, 1/pi, 1/2, 1/2, 1/2, 1/2, ones(1, 13)];  params = fmincon(rmse, scale .\* params0, [], [], [], [], ...  scale .\* lower, scale .\* upper, [], ...  options, scale, fis1, x\_train, y\_train);  fis2 = update\_fis\_params(fis1, params, scale);  scale = ones(1, 21);  params = fmincon(rmse, scale .\* params0, [], [], [], [], ...  scale .\* lower, scale .\* upper, [], ...  options, scale, fis1, x\_train, y\_train);  fis3 = update\_fis\_params(fis1, params, scale);  y1 = reshape(evalfis(fis1, x), length(x1), length(x2))';  rmse1 = sqrt(sum(sum((y\_test - evalfis(fis1, x\_test)).^2)) / numel(y\_test));  print\_surface\_plot(x1, x2, y1, 'Mamdani FIS Before Optmization', ...  'mamdani\_gauss\_5in\_gauss\_5out\_surface\_default.emf', rmse1);  y2 = reshape(evalfis(fis2, x), length(x1), length(x2))';  rmse2 = sqrt(sum(sum((y\_test - evalfis(fis2, x\_test)).^2)) / numel(y\_test));  print\_surface\_plot(x1, x2, y2, 'Mamdani FIS After Optmization With Scale', ...  'mamdani\_gauss\_5in\_gauss\_5out\_surface\_custom.emf', rmse2);  y3 = reshape(evalfis(fis3, x), length(x1), length(x2))';  rmse3 = sqrt(sum(sum((y\_test - evalfis(fis3, x\_test)).^2)) / numel(y\_test));  print\_surface\_plot(x1, x2, y3, 'Mamdani FIS After Optmization Without Scale', ...  'mamdani\_gauss\_5in\_gauss\_5out\_surface\_ custom\_no\_scale.emf', rmse3); |

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



а)

б)

Рисунок — 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 ( и ), 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*