Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Логические основы интеллектуальных систем»

Тема: «Нечеткие нейронные сети»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Кирилович А. А.

**Проверил:**

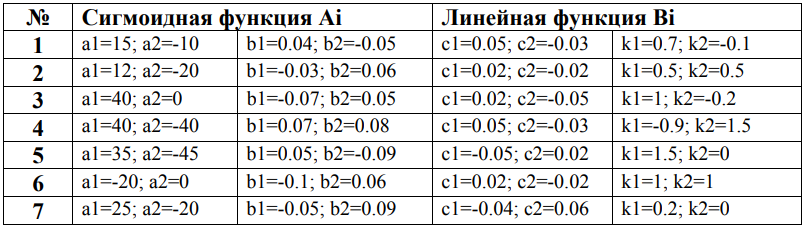
Скарубо А.О.

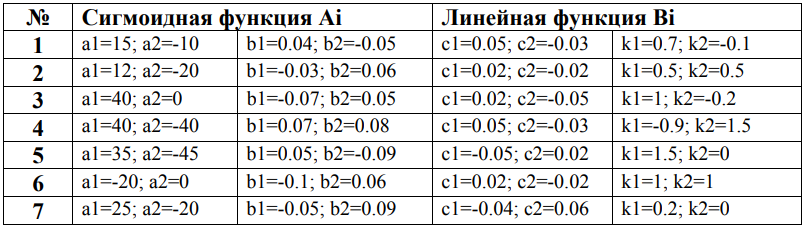
Брест 2023

**Цель работы:** моделирование нечеткого вывода с использованием нейронной сети. Синтез нечетких моделей с помощью системы нейро-нечеткого вывода ANFIS в среде MatLab.

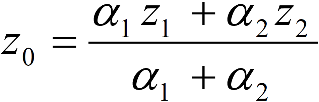
**Ход работы:**

**Вариант 4**





1) Используя эталонную нейро-нечеткую сеть из двух правил, описанную в

теоретической части,

П1: если х есть А1, тогда z есть B1,

П2: если х есть А2, тогда z есть B2,

получить обучающую выборку: множество значений {(x1, z01), . . . , (xP, z0P)} в диапазоне [-100..100], где x – значение входной переменной, а e – четкое значение выхода нечеткой системы.

Неизвестные параметры сигмоидных функции A1 и A2, а так же линейных функции B1 и B2 взять из варианта задания.

2) Используя сформированную выборку в модуле ANFIS системы MatLab

(см. файл anfis\_forms.doc) выполнить построение и обучение (т.е. настройку

параметров ai, bi, ci и ki) нечеткой нейронной сети, рассмотренной в теоретической части Функции принадлежности для входных и выходной переменных подобрать

самостоятельно. Установить оптимальный метод обучения (backpropa или hybrid ).

Привести график промежуточных результатов обучения, а также информацию о числе эпох и значении ошибки в форме таблицы.

3) Протестировать нечеткую нейронную сеть. Сверить результаты работы

обученной нейронной сети с исходными данными (из пункта 1). Результат

представить графически в виде таблицы.

4) Кроме того, для просмотра результатов работы системы воспользоваться

графическими возможностями пакета Fuzzy Logic Toolbox: модуль Rule Viewer и

модуль Surface Viewer .

5) Увеличить количество правил в исходной нейронной сети и повторить

пункты 2-4.

6) Сделать выводы по выполненной работе и оформить отчет.

Вспомогательные функции:

function alpha = sigmFunc(a,b,x)

alpha = 1/(1 + exp(1).^(b.\*(x-a)));

end

function z = linFuncRevers(c,k,a)

z = (a-k)/c;

end

Файл fnn2.m (Нечеткая нейронная сеть для двух правил):

% Задаем параметры нечеткой нейронной сети

a1 = 40; a2 = -40;

b1 = 0.07; b2 = 0.08;

c1 = 0.05; c2 = -0.03;

k1 = -0.9; k2 = 1.5;

% Генерируем обучающую выборку

P\_train = 1000;

X\_train = linspace(-100, 100, P\_train);

Z\_train = zeros(1, P\_train);

for i = 1:P\_train

alpha1 = sigmFunc(a1, b1, X(i));

alpha2 = sigmFunc(a2, b2, X(i));

zet1 = linFuncRevers(c1, k1, alpha1);

zet2 = linFuncRevers(c2, k2, alpha2);

z0 = (alpha1 \* zet1 + alpha2 \* zet2) / (alpha1 + alpha2);

Z\_train(i) = z0;

end

% Генерируем тестовую выборку

P\_test = 200;

X\_test = linspace(-100, 100, P\_test);

Z\_test = zeros(1, P\_test);

for i = 1:P\_test

alpha1 = sigmFunc(a1, b1, X\_test(i));

alpha2 = sigmFunc(a2, b2, X\_test(i));

zet1 = linFuncRevers(c1, k1, alpha1);

zet2 = linFuncRevers(c2, k2, alpha2);

z0 = (alpha1 \* zet1 + alpha2 \* zet2) / (alpha1 + alpha2);

Z\_test(i) = z0;

end

% Строим нечеткую нейронную сеть и обучаем ее

trn\_data = [X\_train', Z\_train'];

val\_data = [X\_test', Z\_test'];

in\_fis = genfis1(trn\_data, 2, 'gbellmf');

% backpropagation

opt\_backprop = anfisOptions('InitialFis', in\_fis, ...

'EpochNumber', 100, ...

'DisplayANFISInformation', 0, ...

'ValidationData', val\_data, ...

'DisplayErrorValues', 0, ...

'OptimizationMethod', 0)

out\_fis\_backprop = anfis(trn\_data, opt\_backprop);

% hybrid

opt\_hybrid = anfisOptions('InitialFis', in\_fis, ...

'EpochNumber', 100, ...

'DisplayANFISInformation', 0, ...

'ValidationData', val\_data, ...

'DisplayErrorValues', 0, ...

'OptimizationMethod', 1);

out\_fis\_hybrid = anfis(trn\_data, opt\_hybrid);

writeFIS(out\_fis\_backprop, 'out\_fis\_backprop.fis');

writeFIS(out\_fis\_hybrid, 'out\_fis\_hybrid.fis');

in\_fis\_backprop = readfis('out\_fis\_backprop.fis');

ruleedit(in\_fis\_backprop);

in\_fis\_hybrid = readfis('out\_fis\_hybrid.fis');

ruleedit(in\_fis\_hybrid);

Результат:

Step size increases to 0.011000 after epoch 5.

Step size increases to 0.012100 after epoch 9.

Step size increases to 0.013310 after epoch 13.

Step size increases to 0.014641 after epoch 17.

Step size increases to 0.016105 after epoch 21.

Step size increases to 0.017716 after epoch 25.

Step size increases to 0.019487 after epoch 29.

Step size increases to 0.021436 after epoch 33.

Step size increases to 0.023579 after epoch 37.

Step size increases to 0.025937 after epoch 41.

Step size increases to 0.028531 after epoch 45.

Step size increases to 0.031384 after epoch 49.

Step size increases to 0.034523 after epoch 53.

Step size increases to 0.037975 after epoch 57.

Step size increases to 0.041772 after epoch 61.

Step size increases to 0.045950 after epoch 65.

Step size increases to 0.050545 after epoch 69.

Step size increases to 0.055599 after epoch 75.

Step size increases to 0.061159 after epoch 81.

Step size decreases to 0.055043 after epoch 86.

Step size decreases to 0.049539 after epoch 90.

Step size decreases to 0.044585 after epoch 94.

Minimal training RMSE = 21.4406

Minimal checking RMSE = 21.4231

Step size increases to 0.011000 after epoch 5.

Step size increases to 0.012100 after epoch 9.

Step size increases to 0.013310 after epoch 13.

Step size increases to 0.014641 after epoch 17.

Step size increases to 0.016105 after epoch 21.

Step size increases to 0.017716 after epoch 25.

Step size increases to 0.019487 after epoch 29.

Step size increases to 0.021436 after epoch 33.

Step size increases to 0.023579 after epoch 37.

Step size increases to 0.025937 after epoch 41.

Step size increases to 0.028531 after epoch 45.

Step size increases to 0.031384 after epoch 49.

Step size increases to 0.034523 after epoch 53.

Step size increases to 0.037975 after epoch 57.

Step size increases to 0.041772 after epoch 61.

Step size increases to 0.045950 after epoch 65.

Step size increases to 0.050545 after epoch 69.

Step size increases to 0.055599 after epoch 73.

Step size increases to 0.061159 after epoch 77.

Step size increases to 0.067275 after epoch 81.

Step size increases to 0.074002 after epoch 85.

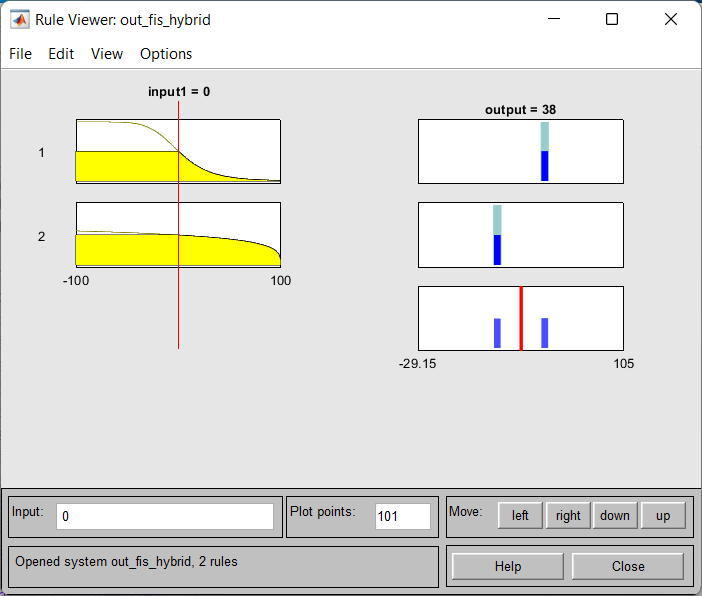
Step size increases to 0.081403 after epoch 89.

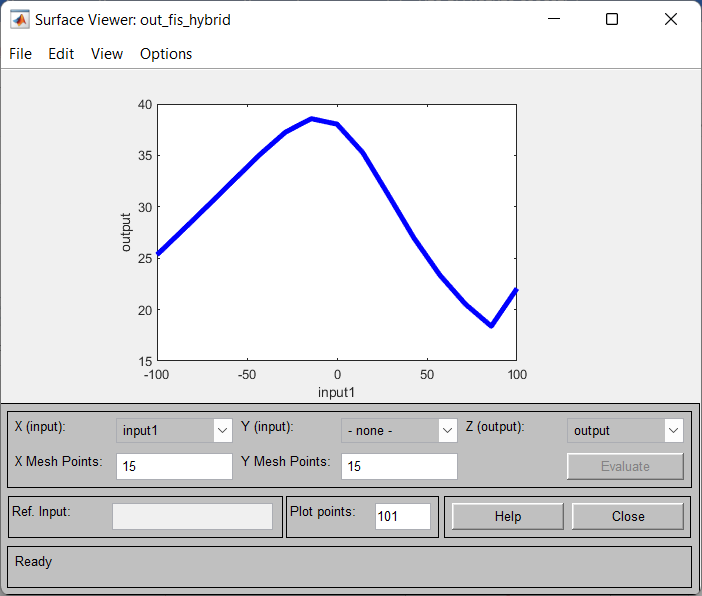
Step size decreases to 0.073262 after epoch 96.

Step size decreases to 0.065936 after epoch 100.

Minimal training RMSE = 0.781782

Minimal checking RMSE = 0.799483





Файл fnn3.m (Нечеткая нейронная сеть для трех правил):

% Задаем параметры нечеткой нейронной сети

a1 = 40; a2 = -40; a3 = 20;

b1 = 0.07; b2 = 0.08; b3 = 0.09;

c1 = 0.05; c2 = -0.03; c3 = -0.04;

k1 = -0.9; k2 = 1.5; k3 = -1.3;

% Генерируем обучающую выборку

P\_train = 1000;

X\_train = linspace(-100, 100, P\_train);

Z\_train = zeros(1, P\_train);

for i = 1:P\_train

alpha1 = sigmFunc(a1, b1, X\_train(i));

alpha2 = sigmFunc(a2, b2, X\_train(i));

alpha3 = sigmFunc(a3, b3, X\_train(i));

zet1 = linFuncRevers(c1, k1, alpha1);

zet2 = linFuncRevers(c2, k2, alpha2);

zet3 = linFuncRevers(c3, k3, alpha3);

z0 = (alpha1 \* zet1 + alpha2 \* zet2 + alpha3 \* zet3) / (alpha1 + alpha2 + alpha3);

Z\_train(i) = z0;

end

% Генерируем тестовую выборку

P\_test = 200;

X\_test = linspace(-100, 100, P\_test);

Z\_test = zeros(1, P\_test);

for i = 1:P\_test

alpha1 = sigmFunc(a1, b1, X\_test(i));

alpha2 = sigmFunc(a2, b2, X\_test(i));

alpha3 = sigmFunc(a3, b3, X\_test(i));

zet1 = linFuncRevers(c1, k1, alpha1);

zet2 = linFuncRevers(c2, k2, alpha2);

zet3 = linFuncRevers(c3, k3, alpha3);

z0 = (alpha1 \* zet1 + alpha2 \* zet2 + alpha3 \* zet3) / (alpha1 + alpha2 + alpha3);

Z\_test(i) = z0;

end

% Строим нечеткую нейронную сеть и обучаем ее

trn\_data = [X\_train', Z\_train'];

val\_data = [X\_test', Z\_test'];

in\_fis = genfis1(trn\_data, 3, 'gbellmf');

% backpropagation

opt\_backprop = anfisOptions('InitialFis', in\_fis, ...

'EpochNumber', 100, ...

'DisplayANFISInformation', 0, ...

'ValidationData', val\_data, ...

'DisplayErrorValues', 0, ...

'OptimizationMethod', 0);

out\_fis\_backprop = anfis(trn\_data, opt\_backprop);

% hybrid

opt\_hybrid = anfisOptions('InitialFis', in\_fis, ...

'EpochNumber', 100, ...

'DisplayANFISInformation', 0, ...

'ValidationData', val\_data, ...

'DisplayErrorValues', 0, ...

'OptimizationMethod', 1);

out\_fis\_hybrid = anfis(trn\_data, opt\_hybrid);

Результат:

Step size increases to 0.011000 after epoch 5.

Step size increases to 0.012100 after epoch 9.

Step size increases to 0.013310 after epoch 13.

Step size increases to 0.014641 after epoch 17.

Step size increases to 0.016105 after epoch 21.

Step size increases to 0.017716 after epoch 25.

Step size increases to 0.019487 after epoch 29.

Step size decreases to 0.017538 after epoch 35.

Step size decreases to 0.015785 after epoch 39.

Step size decreases to 0.014206 after epoch 43.

Step size decreases to 0.012786 after epoch 47.

Step size decreases to 0.011507 after epoch 51.

Step size decreases to 0.010356 after epoch 55.

Step size decreases to 0.009321 after epoch 59.

Step size decreases to 0.008389 after epoch 63.

Step size decreases to 0.007550 after epoch 67.

Step size decreases to 0.006795 after epoch 71.

Step size decreases to 0.006115 after epoch 75.

Step size decreases to 0.005504 after epoch 79.

Step size decreases to 0.004953 after epoch 83.

Step size decreases to 0.004458 after epoch 87.

Step size decreases to 0.004012 after epoch 91.

Step size decreases to 0.003611 after epoch 95.

Step size decreases to 0.003250 after epoch 99.

Minimal training RMSE = 3.25523

Minimal checking RMSE = 3.25554

Step size increases to 0.011000 after epoch 5.

Step size increases to 0.012100 after epoch 9.

Step size increases to 0.013310 after epoch 13.

Step size increases to 0.014641 after epoch 17.

Step size increases to 0.016105 after epoch 21.

Step size increases to 0.017716 after epoch 25.

Step size increases to 0.019487 after epoch 29.

Step size increases to 0.021436 after epoch 33.

Step size increases to 0.023579 after epoch 37.

Step size increases to 0.025937 after epoch 41.

Step size increases to 0.028531 after epoch 45.

Step size increases to 0.031384 after epoch 49.

Step size increases to 0.034523 after epoch 53.

Step size increases to 0.037975 after epoch 57.

Step size increases to 0.041772 after epoch 61.

Step size increases to 0.045950 after epoch 65.

Step size increases to 0.050545 after epoch 69.

Step size increases to 0.055599 after epoch 73.

Step size increases to 0.061159 after epoch 77.

Step size increases to 0.067275 after epoch 81.

Step size increases to 0.074002 after epoch 85.

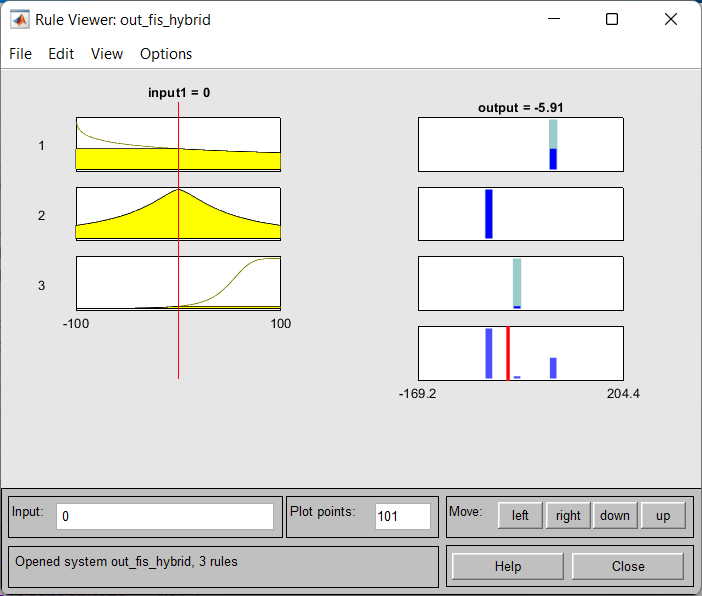
Step size increases to 0.081403 after epoch 89.

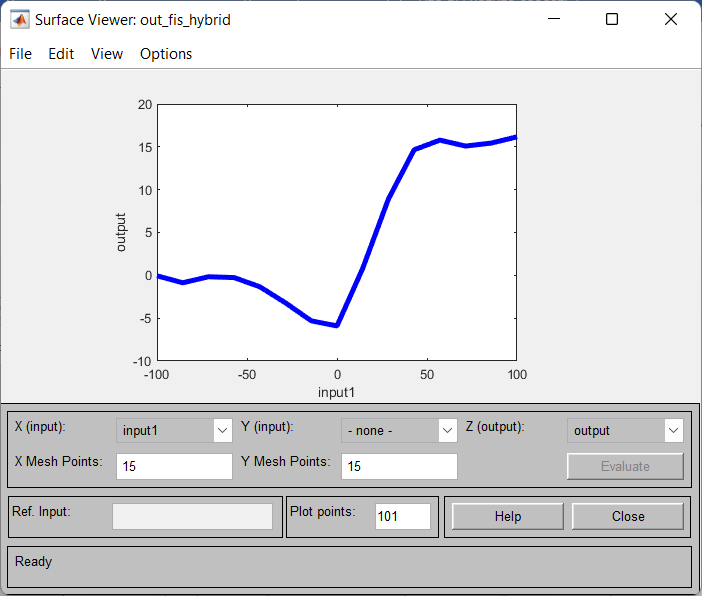
Step size increases to 0.089543 after epoch 93.

Step size decreases to 0.080589 after epoch 99.

Minimal training RMSE = 0.324678

Minimal checking RMSE = 0.326842





**Вывод**: осуществил моделирование нечеткого вывода с использованием нейронной сети. Синтез нечетких моделей с помощью системы нейро-нечеткого вывода ANFIS в среде MatLab.