

Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт з дисципліни
**«НЕЧІТКІ МОДЕЛІ І МЕТОДИ
ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ»**
для студентів спеціальності
8.05010104 “Системи штучного інтелекту”
усіх форм навчання

2015

Методичні вказівки до виконання лабораторних робіт з дисципліни "Нечіткі моделі і методи обчислювального інтелекту" для студентів спеціальності 8.05010102 "Системи штучного інтелекту" усіх форм навчання / Уклад.: С.О. Субботін. – Запоріжжя: ЗНТУ, 2015. – 50 с.

Автор: Сергій Олександрович Субботін,
професор кафедри програмних засобів,
доктор технічних наук, професор

Рецензент: А.О. Олійник, к.т.н., доцент

Відповідальний
за випуск: В.І. Дубровін, зав. каф. "Програмні засоби"

Затверджено
на засіданні кафедри
"Програмні засоби"

Протокол № 11
від 04.06.2013 р.

ЗМІСТ

Загальні положення	4
1 Лабораторна робота № 1.	5
2 Лабораторна робота № 2.	8
3 Лабораторна робота № 3.	10
Література	12
Додаток А. Приклади виконання елементів завдань.	14
Додаток Б. Нечітка логіка та нейро-нечіткі мережі у пакеті MATLAB.	32

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання нечітких методів і моделей для побудови інтелектуальних систем.

Відповідно до графіка студенти перед виконанням лабораторної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою.

Для одержання заліку по кожній роботі студент здає викладачу цілком оформлений звіт, а також CD-RW (DVD-RW) диск у форматі MS – DOS / Windows, перевірений на відсутність вірусів, з текстами розроблених програм, файлами програм, що виконуються, файлами даних і текстом звіту.

Звіт (приклад оформлення - див. додаток А) має містити:

- титульний аркуш (на ньому вказують назву міністерства, назву університету, назву кафедри, номер, вид і тему роботи, виконавця та особу, що приймає звіт, рік);

- мету, варіант і завдання роботи;
- лаконічний опис теоретичних відомостей;
- текст програми, що обов'язково містить коментарі;
- вхідні та вихідні дані програми;
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word 97: шрифт Times New Roman, 12 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм, таблиць і роздруківок даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах.

Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці опису відповідної роботи.

Загальний залік студент одержує після виконання і здачі останньої роботи.

1 Лабораторна робота №1

НЕЧІТКІ МНОЖИНИ ТА ОПЕРАЦІЇ З НИМИ У ПАКЕТІ MATLAB

Мета роботи: Освоїти основні поняття нечіткої логіки та навчитися працювати із нечіткими множинами.

Завдання до роботи

1. Ознайомитися з конспектом лекцій, рекомендованою літературою та теоретичними відомостями даних методичних вказівок.
2. Згідно з номером студента за журналом для варіанту з відповідним номером виконати індивідуальне завдання.
3. Оформити звіт з роботи.
4. Відповісти на контрольні запитання.

Індивідуальні завдання

Нехай V - номер варіанта студента за журналом. Визначимо:

$$V_1 = \begin{cases} 4V, & \text{якщо } V < 5; \\ V, & \text{якщо } 5 \leq V \leq 10; \\ \text{round}(0,3V), & \text{якщо } V > 10, \end{cases}$$

де $\text{round}(x)$ - функція округлення.

Сформуємо нечіткі множини:

$$A = \{ \langle \frac{k}{V_1} \mid k \rangle \},$$

$$B = \{ \langle \frac{2k}{V_1} \mid 0,5k \rangle \},$$

$$k = 0, 1, \dots, V.$$

Завдання 1. Побудувати декілька разів графіки функцій приналежності у залежності від номеру варіанту із різними значеннями параметрів. Параметри функцій приналежності задати самостійно. Дослідити, як впливають значення параметрів на зміну значення функцій приналежності. Завдання виконати у пакеті MATLAB із використанням функцій модуля Fuzzy Logic Toolbox.

V	Функції		V	Функції	
1	dsigmf	gbellmf	2	psigmf	gaussmf
3	dsigmf	zmf	4	psigmf	pimf
5	dsigmf	trimf	6	sigmf	dsigmf
7	gauss2mf	gbellmf	8	sigmf	pimf
9	gauss2mf	trimf	10	sigmf	trimf
11	gauss2mf	trapmf	12	smf	gauss2mf
13	gaussmf	gauss2mf	14	smf	gbellmf
15	gaussmf	pimf	16	trapmf	gaussmf
17	gaussmf	trapmf	18	trimf	gauss2mf
19	gbellmf	pimf	20	trimf	gbellmf
21	gbellmf	smf	22	zmf	smf
23	pimf	gaussmf	24	zmf	dsigmf
25	pimf	sigmf	26	zmf	pimf

Завдання 2. Визначити значення елементів нечітких множин А та В.
Завдання виконати вручну.

Завдання 3. Для кожної з нечітких множин А та В визначити: висоту, чи є множина нормальною, субнормальною або порожньою; чи є множина унімодальною, ядро, носій, межі, точки переходу, найближчу чітку множину, чи є множина опуклою, міру нечіткості Егера (у метриках: $p=1$ та $p=2$), міру нечіткості Коско, чітку множину альфа-рівня. Завдання виконати вручну.

Завдання 4. Визначити результати виконання над нечіткими множинами А та В операцій: доповнення, включення, рівність, об'єднання, перетинання, різниця, симетрична різниця, диз'юнктивна сума, алгебраїчний добуток, алгебраїчна сума, обмежена сума, обмежена різниця, обмежений добуток, драстичне перетинання, драстичне об'єднання, λ -сума, зведення в ступінь, CON, DIL, множення на число, опукла комбінація, нормалізація, нечітке включення. У якості скалярного числа де потрібно використовувати число V_1 або V_1^{-1} . Завдання виконати вручну.

Завдання 5. Придумати п'ять лінгвістичної змінної та варіантів її значень. Для лінгвістичної змінної запропонувати нечіткі змінні. Навести приклади можливих значень нечітких змінних. Визначити терм-множину лінгвістичної змінної.

Завдання 6. Для множин А та В за допомогою пакету MATLAB визначити результати нечітких операцій: 'sum' - сума, 'sub' - вирахування, 'prod' - добуток, 'div' - ділення. Результати зобразити графічно у пакеті MATLAB. При виконанні завдання використовувати функції приналежності з завдання 1.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості.
3. Номер варіанту та завдання до роботи.
4. Опис процесу та результати виконання завдань.
5. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

Контрольні запитання

1. Де і для яких задач доцільно застосовувати нечітку логіку?
2. Що таке нечітка множина?
3. Дайте визначення та наведіть приклади понять: чітка підмножина, функція приналежності, нечітка підмножина, множина приналежностей, нечітка змінна, лінгвістична змінна, терм-множина, терм, семантична процедура, нечітке число, нечітке число (L-R)-типу, функція приналежності нечітких чисел (L-R)-типу, нечітке n -арне відношення, нечітке відношення на множині.
4. У чому відмінності звичайних та нечітких чисел?
5. Характеристики та властивості нечітких множин: висота, чи є множина нормальною, субнормальною або порожньою; чи є множина унімодальною, ядро, носій, межі, точки переходу, найближча чітку множину, чи є множина опуклою, міра нечіткості Егера, міру нечіткості Коско, чітка множина альфа-рівня.
6. Операції над нечіткими множинами: доповнення, включення, рівність, об'єднання, перетинання, різниця, симетрична різниця, диз'юнктивна сума, алгебраїчний добуток, алгебраїчна сума, обмежена сума, обмежена різниця, обмежений добуток, драстичне перетинання, драстичне об'єднання, λ -сума, зведення в ступінь, CON, DIL, множення на число, опукла комбінація, декартовий добуток, нормалізація, нечітке включення.
7. Властивості нечітких операцій та закони нечіткої логіки.
8. Функції приналежності нечітких множин: аналітичний запис, формат виклику у пакеті MATLAB, графіки.
9. Нечіткі операції у пакеті MATLAB.

2 Лабораторна робота № 2

МОДЕЛІ НА БАЗІ ТЕОРІЇ НЕЧІТКИХ МНОЖИН

Мета роботи: Освоїти основні принципи побудови нейро-нечітких мереж у пакеті MATLAB.

Завдання до роботи

1. Ознайомитися з конспектом лекцій, рекомендованою літературою та теоретичними відомостями даних методичних вказівок.
2. Згідно з номером студента за журналом для варіанту з відповідним номером виконати індивідуальне завдання.
3. Оформити звіт з роботи.
4. Відповісти на контрольні запитання.

Індивідуальні завдання

Завдання 1. Придумати приклад нечіткого відношення $A \ R \ B$, а також його доповнення $\bar{R}$. Визначити результати операцій:

$$C = R \cup \bar{R}, D = R \cap \bar{R}, E = (C \cap D) \cup R.$$

Завдання виконати вручну.

Завдання 2. Для нечітких множин A та B , отриманих у попередній роботі відповідно до V - номера варіанта студента за журналом, знайти чіткі значення A^* та B^* , використовуючи методи дефазифікації: середній з максимальних, найбільший з максимальних, найменший з максимальних, центр тяжіння, метод медіани, висотна дефазифікація. Порівняти результати методів для кожної нечіткої множини. Завдання виконати вручну.

Завдання 3. Використовуючи продукційну модель бази знань експертної системи, розробленої у попередньому семестрі, побудувати її спрощений варіант у вигляді правил "Якщо-то". Використовувати не більше чотирьох вхідних змінних та п'яти значень вихідної змінної. Задати не більше вісьми правил. У редакторі FIS пакету MATLAB побудувати систему нечіткого виведення для створених змінних і правил. Переглянути результати роботи FIS для позитивного та негативного сценаріїв. Зберегти створену FIS у файл на диску.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості.
3. Номер варіанту та завдання до роботи.
4. Опис процесу та результати виконання завдань. Зображення екранних форм, текст створеної FIS.
5. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

Контрольні запитання

1. Інженерія знань та нечіткість.
2. Нечіткі відношення та операції з ними.
3. Нечіткі множини та виведення.
4. Дефузифікація: загальне призначення та методи: середній з максимальних, найбільший з максимальних, найменший з максимальних, центр тяжіння, метод медіани, висотна дефазифікація.
5. Методи нечіткого виведення.
6. Порівняння методів побудови нечіткого логічного виведення Мамдані та Сугено.
7. Створення нечітких моделей у пакеті MATLAB.

3 Лабораторна робота № 3

ПОБУДОВА НЕЙРО-НЕЧІТКИХ МЕРЕЖ У ПАКЕТІ MATLAB

Мета роботи: Освоїти основні принципи побудови нейро-нечітких мереж у пакеті MATLAB.

Завдання до роботи

1. Згідно з номером індивідуального варіанта студента за журналом згенерувати навчаючу та тестову вибірки даних. Нехай V - номер студента за журналом. Визначимо кількість екземплярів у навчальній вибірці S_n , кількість екземплярів у тестовій вибірці S_t , кількість вхідних змінних (ознак) вибірок N , значення ознак вибірок: навчальної - x_n та тестової - x_t , а також значення цільових ознак для вибірок : навчальної - y_n та тестової - y_t .

$$N = \begin{cases} V, & \text{якщо } V < 15, \\ V - 14, & \text{якщо } V > 15; \end{cases}$$

$$S_n = \begin{cases} 10V, & \text{якщо } V < 5, \\ 5V, & \text{якщо } 5 \leq V < 10, \\ 2V, & \text{якщо } 10 \leq V < 20, \\ V, & \text{інакше;} \end{cases} \quad S_t = \begin{cases} 8V, & \text{якщо } V < 5, \\ 6V, & \text{якщо } 5 \leq V < 10, \\ 3V, & \text{якщо } 10 \leq V < 20, \\ 2V, & \text{інакше;} \end{cases}$$

$$x_n = \{x_j^s\}, y_n = \{y^s\}, s=1,2,\dots,S_n;$$

$$x_t = \{x_j^s\}, y_t = \{y^s\}, s=1,2,\dots,S_t; ,$$

$$x_j^s = \frac{V}{s \cdot j} \cdot \text{rand}, j=1,2,\dots,N, y^s = \begin{cases} 0,1(x^{s_1} + x^{s_2}), & x^{s_1} < V, \\ 0,3(x^{s_1} - x^{s_2}), & x^{s_1} > V, \\ 0,5x^{s_1}, & x^{s_1} = V. \end{cases}$$

Тут rand - функція, що генерує псевдовипадкові числа у діапазоні $[0; 1]$.

2. Для згенерованих вибірок за допомогою редактору `anfisedit` побудувати нейро-нечітку модель. Спробувати використати різні алгоритми кластеризації, різні кількості функцій приналежності для входів, різні кількості циклів навчання та різні алгоритми навчання.

3. Протестувати побудовану модель.

4. Проаналізувати отримані результати та відповісти на питання: який алгоритм кластер-аналізу призводить до отримання мережі меншої складності; як впливає задана кількість циклів навчання на точність навчання; як впливає задана точність навчання на тривалість навчання; які вимоги мають пред'являтися до навчальної вибірки та як це вплине на процес навчання.

5. Виконати пункти 1-4 у режимі командного вікна без застосування редактору `anfisedit`, використовуючі функції модуля `Fuzzy Logic Toolbox`.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості.
3. Номер варіанту та завдання до роботи.
4. Опис процесу та результати виконання завдань. Тексти програм. Згенеровані вибірки. Зображення структури отриманої мережі.
5. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

Контрольні запитання

1. Структура та елементи нейро-нечітких мереж.
2. Застосування нейро-нечітких мереж для видобування знань з даних.
8. Нечітка кластеризація як підхід до представлення знань.
3. Який алгоритм кластер-аналізу призводить до отримання мережі меншої складності?
4. Як впливає задана кількість циклів навчання на точність навчання?
5. Як впливає задана точність навчання на тривалість навчання?
6. Які вимоги мають пред'являтися до навчальної вибірки та як це вплине на процес навчання?
7. Функції пакету `MATLAB` для створення нейро-нечітких мереж.
8. Редактор `anfisedit`.

ЛІТЕРАТУРА

Основна література

1. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навчальний посібник. – Запоріжжя: ЗНТУ, 2008. – 341 с.
2. Леоненков А.В. Нечеткое моделирование в среде MATLAB. – СПб.: БХВ – Петербург, 2003. – 736 с.

Додаткова література

3. Алиев Р.А., Абдикеев Н.М., Шахназаров М.М. Производственные системы с искусственным интеллектом. – М: Радио и связь, 1990. – 264 с.
4. Архангельский В.И., Богаенко И.Н., Грабовский Г.Г., Рюмшин Н.А. Системы фуцци – управления. – К.: Тэхника, 1997. – 208 с.
5. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях – В сб.: Вопросы анализа и процедуры принятия решений. – М: Мир, 1976. – С.172 – 215.
6. Бокша В.В., Силов В.Б. Нечеткое целевое управление системами с заданным конечным состоянием // Автоматика. – 1985. – № 3. – С.3 – 8.
7. Борисов А.Н., Алексеев А.В., Крумберг О.А. и др. Модели принятия решений на основе лингвистической переменной. – Рига: Зинатне, 1982. – 256 с.
8. Борисов А.Н., Алексеев А.В., Меркурьева Г.В. и др. Обработка нечеткой информации в системах принятия решений. – М: Радио и связь. 1989. – 304 с.
9. Борисов А.Н., Крумберг О.А., Федоров И.П. Принятие решений на основе нечетких моделей. Примеры использования. – Рига:Зинатне, 1990. – 184 с.
10. Гусев Л.А., Смирнова И.М. Размытые множества. Теория и приложения (обзор) // Автоматика и телемеханика. – 1973. – № 5. – С.66 – 85.
11. Дьяконов В. MATLAB 6: учебный курс. – СПб.: Питер, 2001. – 592 с.
12. Заде Л.А. Основы нового подхода к анализу сложных систем и процессов принятия решений. – В кн.: Математика сегодня. – М.:Знание, 1974. – С. 5 – 49.
13. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.:Мир, 1976. – 165 с.
14. Заде Л.А. Размытые множества и их применение в распознавании образов и кластер – анализе. – В сб.: Классификация и кластер. – М: Мир, 1980, с.208 – 247.

15. Зуенков М.А. Приближение характеристических функций нечетких множеств // Автоматика и телемеханика. – 1984. – № 10. – С.138 – 149.
16. Кандель А., Байатт У.Дж. Нечеткие множества, нечеткая алгебра, нечеткая статистика // Труды американского общества инженеров – радиоэлектроников. – 1978. – Т. 66. – №12. – С.37 – 61.
17. Кофман А. Введение в теорию нечетких множеств. – М.: Радио и связь, 1982. – 432 с.
18. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382с.
19. Малышев Н.Г., Бернштейн Л.С., Боженюк А.В. Нечеткие модели для экспертных систем в САПР. — М.: Энергоиздат, 1991. — 136 с.
20. Мелихов А.Н., Берштейн Л.С., Коровин С.Я. Ситуационные советующие системы с нечеткой логикой. – М.: Наука, 1990. – 272 с.
21. Мешалкин В.П. Экспертные системы в химической технологии. – М.: Химия, 1995. – 368 с.
22. Несенюк А.П. Неопределенные величины в задачах управления с неполной информацией // Автоматика. – 1979. – № 2. – С.55 – 64.
23. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. – М.: Наука, 1986. – 312 с.
24. Нечеткие множества в моделях управления и искусственного интеллекта/А.Н. Аверкин, И.З. Батыршин, А.Ф. Блишун, В.Б. Силов, В.Б. Тарасов. Под ред. Д.А. Поспелова. – М.:Наука, 1986. – 312 с.
25. Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – 408 с.
26. Орлов А.И. Задачи оптимизации и нечеткие переменные. – М.: Знание, 1980. – 64 с.
27. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. – М.: Радио и связь, 1981. – 286 с.
28. Осуга С. Обработка знаний. – М.: Мир, 1989. – 293 с.
29. Поспелов Д.А. Логико – лингвистические модели в системах управления. – М.:Энергоиздат, 1981. – 232 с.
30. Прикладные нечеткие системы / Асаи К., Ватада Д., Иваи С. и др./Под ред. Т. Тэрано, К. Асаи, М. Сугено. – М.: Мир, 1993. – 368 с.
31. Приобретение знаний / Под ред. С. Осуги, Ю. Саэки. – М.: Мир, 1990. – 304 с.
32. Рідкокаша А.А., Голдер К.К. Основи систем штучного інтелекту. Навчальний посібник. Черкаси, "ВІДЛІУННЯ – ПЛЮС", 2002. – 240 с.

Додаток А
Приклади виконання елементів завдань

Міністерство освіти і науки України
Запорізький національний технічний університет
Інститут інформатики та радіоелектроніки

кафедра програмних засобів

ЗВІТ
з лабораторної роботи № 1
з дисципліни
"ОСНОВИ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ"
на тему:
"НЕЧІТКІ МНОЖИНИ ТА ОПЕРАЦІЇ З НИМИ
У ПАКЕТІ MATLAB"

Виконав: ст. гр. ІОТ-413

_____ А.М.Іванов

Прийняв: к.т.н., доцент

_____ С.О. Субботін

Запоріжжя
2013

А.1 Приклади запису нечітких множин

Приклад 1. Нехай $E = \{x_1, x_2, x_3, x_4, x_5\}$, $M = [0, 1]$; A - нечітка множина, для якої $\mu_A(x_1) = 0,3$; $\mu_A(x_2) = 0$; $\mu_A(x_3) = 1$; $\mu_A(x_4) = 0,5$; $\mu_A(x_5) = 0,9$. Тоді A можна представити у вигляді: $A = \{0,3/x_1; 0/x_2; 1/x_3; 0,5/x_4; 0,9/x_5\}$ або $A = \{0,3/x_1; 0/x_2; 1/x_3; 0,5/x_4; 0,9/x_5\}$ або $A = 0,3/x_1 + 0/x_2 + 1/x_3 + 0,5/x_4 + 0,9/x_5$ або $A = \{0,3/x_1; 0/x_2; 1/x_3; 0,5/x_4; 0,9/x_5\}$ або $A = \{<0,3/x_1>, <0/x_2>, <1/x_3>, <0,5/x_4>, <0,9/x_5>\}$ або

$$A = \begin{array}{c|c|c|c|c} x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline 0,3 & 0 & 1 & 0,5 & 0,9 \end{array} \text{ або } A = \begin{array}{c|c|c|c|c} 0,3 & 0 & 1 & 0,5 & 0,9 \\ \hline x_1 & x_2 & x_3 & x_4 & x_5 \end{array}$$

Тут знак «+» не є позначенням операції додавання, а має сенс об'єднання.

Приклад 2. Нехай $E = \{0, 1, 2, \dots, 10\}$, $M = [0, 1]$. Нечітку множину «декілька» можна визначити в такий спосіб: «декілька» = $0,5|3 + 0,8|4 + 1|5 + 1|6 + 0,8|7 + 0,5|8$; її характеристики: висота = 1, носій = $\{3, 4, 5, 6, 7, 8\}$, точки переходу - $\{3, 8\}$.

Приклад 3. Нехай $E = \{0, 1, 2, 3, \dots, n, \dots\}$. Нечітку множина «малий» можна визначити:

$$\text{малий} = \mu_{\text{малий}}(n) = \frac{1}{n} \left(\frac{1}{1 + \ln n} \right).$$

Приклад 4. Нехай $E = \{1, 2, 3, \dots, 100\}$ і відповідає поняттю «вік», тоді нечітка множина «молодий», може бути визначене в такий спосіб:

$$\mu_{\text{молодий}}(x) = \begin{cases} 1, & x \in [1, 25] \\ \frac{1}{1 + \ln(x - 25)}, & x > 25. \end{cases}$$

Нечітка множина «молодий» на універсальній множині $E' = \{\text{Іванов, Петров, Сидорів, ...}\}$ задається за допомогою функції приналежності $\mu_{\text{молодий}}(x)$ на $E = \{1, 2, 3, \dots, 100\}$ (вік), названої стосовно E' функцією сумісності, при цьому: $\mu_{\text{молодий}}(\text{Сидорів}) = \mu_{\text{молодий}}(x)$, де x - вік Сидорова.

Приклад 5. Нехай E - множина цілих чисел: $E = \{-8, -5, -3, 0, 1, 2, 4, 6, 9\}$. Тоді нечітку підмножину чисел, за абсолютною величиною близьких до нуля можна визначити, наприклад, так: $A = \{0/-8 + 0,5/-5 + 0,6/-3 + 1/0 + 0,9/1 + 0,8/2 + 0,6/4 + 0,3/6 + 0/9\}$.

А.2 Приклади визначення характеристик нечітких множин та результатів операцій над нечіткими множинами

Приклад 6. Нехай ми маємо нечіткі множини А та В. Визначимо їхні властивості. У якості скалярного значення будемо використовувати число $V_1=3$, або, за потреби, $V_1^{-1} \approx 0,3$.

назва множини	А	В
елементи множини	{0,1 1; 0,5 2; 1,0 3; 0,8 3,2}	{0,6 1; 0,1 2; 0,1 3}
характеристики:		
висота	1,0	0,6
нормальність	нормальна	субнормальна
порожність	непорожня	непорожня
унімодальність	унімодальна	неунімодальна
ядро	{3}	$\emptyset$
носій	{1; 2; 3; 3,2}	{1; 2; 3}
межі	{0,1 1; 0,5 2; 0,8 3,2}	{0,6 1; 0,1 2; 0,1 3}
точки переходу	{0,5 2}	$\emptyset$
найближча чітка множина	{2; 3; 3,2}	{1}
опуклість	опукла	опукла
міра нечіткості Егера в метриці $p=1$	0,4	0,4
міра нечіткості Егера в метриці $p=2$	0,5	0,337
міра нечіткості Коско	0,348	0,25
чітка множина альфа-рівня, $\alpha=V_1^{-1}=0,3$	{2; 3; 3,2}	{1}
операції:		
доповнення	{0,9 1; 0,5 2; 0,2 3,2}	{0,4 1; 0,9 2; 0,9 3}
включення	$A \subseteq B$	
рівність	$A \neq B$	
об'єднання	{0,6 1; 0,5 2; 1,0 3; 0,8 3,2}	
перетинання	{0,1 1; 0,1 2; 0,1 3}	
різниця	{0,4 2; 0,9 3; 0,8 3,2}	

назва множини	A		B
симетрична різниця	{0,5 1; 0,4 2; 0,9 3; 0,8 3,2}		
диз'юнктивна сума	{0,6 1; 0,5 2; 0,9 3}		
алгебраїчний добуток	{0,06 1; 0,05 2; 0,1 3}		
алгебраїчна сума	{0,694 1; 0,595 2; 1,0 3; 0,8 3,2}		
обмежена сума	{0,7 1; 0,6 2; 1,0 3; 0,8 3,2}		
обмежена різниця	{0,4 2; 0,9 3; 0,8 3,2}		
обмежений добуток	{0,1 3}		
драстичне перетинання	{0,1 3}		
драстичне об'єднання	{1,0 1; 1,0 2; 1,0 3; 0,8 3,2}		
λ -сума, $\lambda = V_1^{-1}=0,3$	{0,45 1; 0,22 2; 0,37 3; 0,24 3,2}		
зведення в ступінь, $V_1=3$	{0,001 1; 0,125 2; 1,0 3; 0,512 3,2}	{0,216 1; 0,001 2; 0,001 3}	
CON	{0,01 1; 0,25 2; 1,0 3; 0,64 3,2}	{0,36 1; 0,01 2; 0,01 3}	
DIL	{0,316 1; 0,707 2; 1,0 3; 0,894 3,2}	{0,775 1; 0,316 2; 0,316 3}	
множення на число, $a=V_1^{-1}=0,3$	{0,3 1; 0,15 2; 0,3 3; 0,24 3,2}	{0,18 1; 0,03 2; 0,03 3}	
опукла комбінація, $\lambda_1=V_1^{-1}=0,3$, $\lambda_2=1-\lambda_1=0,7$	{0,72 1; 0,22 2; 0,37 3; 0,24 3,2}		
нормалізація	{0,1 1; 0,5 2; 1,0 3; 0,8 3,2}	{1,0 1; 0,167 2; 0,167 3}	
нечітке включення за Лукасевичем	{0,1 3}		
нечітке включення за Заде	{0,1 3}		

Приклад 7. Нехай $E = \{1, 2, 3, 4\}$; $A = 0,8/1 + 0,6/2 + 0/3 + 0/4$; $K(1) = 1/1 + 0,4/2$; $K(2) = 1/2 + 0,4/1 + 0,4/3$; $K(3) = 1/3 + 0,5/4$; $K(4) = 1/4$. Тоді $H(A, K) = \mu_A(1)K(1) \cup \mu_A(2)K(2) \cup \mu_A(3)K(3) \cup \mu_A(4)K(4) = 0,8(1/1 + 0,4/2) \cup 0,6(1/2 + 0,4/1 + 0,4/3) = 0,8/1 + 0,6/2 + 0,24/3$.

А.3 Приклади визначення нечітких відношень

Приклад 8. Нехай $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3, y_4\}$, $M = [0, 1]$. Нечітке відношення $R = X R Y$ може бути задане таблицею:

R	y ₁	y ₂	y ₃	y ₄
x ₁	0	0	0.1	0.3
x ₂	0	0.8	1	0.7
x ₃	1	0.5	0.6	1

Приклад 9. Нехай $X = Y = (-\infty, \infty)$, тобто множина усіх дійсних чисел. Відношення $x \gg y$ (x забагато більший ніж y) можна задати функцією приналежності:

$$\mu_R = \begin{cases} 0, & \text{якщо } x \leq y, \\ \frac{1}{1 + \frac{1}{(x-y)^2}}, & \text{якщо } y > x. \end{cases}$$

Приклад 10. Відношення R , для котрого $\mu_R(x, y) = e^{-k(x-y)^2}$ при достатньо великих k можна інтерпретувати так: « x та y близькі одне до одного числа».

Приклад 11. Нехай нечіткі відношення R_1 та R_2 задаються таким таблицями:

R ₁	y ₁	y ₂	y ₃
x ₁	0.1	0.7	0.4
x ₂	1	0.5	0

R ₂	z ₁	z ₂	z ₃	z ₄
y ₁	0.9	0	1	0.2
y ₂	0.3	0.6	0	0.9
y ₃	0.1	1	0	0.5

Тоді нечітке відношення $R_1 \bullet R_2$ може бути задане таблицею:

R ₁ •R ₂	z ₁	z ₂	z ₃	z ₄
x ₁	0.3	0.6	0.1	0.7
x ₂	0.9	0.5	1	0.5

При цьому:

$$\begin{aligned}\mu_{R_1 \bullet R_2}(x_1, z_1) &= [\mu_{R_1}(x_1, y_1) \cap \mu_{R_2}(y_1, z_1)] \cup [\mu_{R_1}(x_1, y_2) \cap \mu_{R_2}(y_2, z_1)] \cup \\ &\cup [\mu_{R_1}(x_1, y_3) \cap \mu_{R_2}(y_3, z_1)] = (0.1 \cap 0.9) \cup (0.7 \cap 0.3) \cup (0.4 \cap 0.1) = \\ &= 0.1 \cup 0.3 \cup 0.1 = 0.3 \\ \mu_{R_1 \bullet R_2}(x_1, z_2) &= (0.1 \cap 0) \cup (0.7 \cap 0.6) \cup (0.4 \cap 1) = 0 \cup 0.6 \cup 0.4 = 0.6\end{aligned}$$

У цьому прикладі спочатку використаний «аналітичний» спосіб композиції відносин R_1 і R_2 , тобто i -й рядок R_1 «збільшується» на j -й стовпець R_2 з використанням операції $\cap$, потім отриманий результат «згортається» з використанням операції $\cup$ в $\mu(x_i, z_j)$.

А.4 Приклад дефазифікації

Приклад 12. Нехай ми маємо нечітку множину для поняття "чоловік середнього росту": $A = \{0|155, 0,1|160, 0,3|165, 0,8|170, 1|175, 1|180, 0,5|185, 0|190\}$. Необхідно провести дефазифікацію нечіткої множини "чоловік середнього росту" за методом центра тяжіння.

Застосуємо формулу:

$$a = \frac{\sum_{i=1}^k x_i \mu_A(x_i)}{\sum_{i=1}^k \mu_A(x_i)},$$

де k - кількість елементів у множині A , та отримаємо:

$$a = (0 \cdot 155 + 0,1 \cdot 160 + 0,3 \cdot 165 + 0,8 \cdot 170 + 1 \cdot 175 + 1 \cdot 180 + 0,5 \cdot 185 + 0 \cdot 190) / (0 + 0,1 + 0,3 + 0,8 + 1 + 1 + 0,5 + 0) = 175,4.$$

А.5 Приклад спадного нечіткого виведення

Приклад 13. Спадне нечітке логічне виведення. Нехай задана модель діагностики системи, що складається з двох передумов і трьох висновків: $X = \{x_1, x_2\}$, $Y = \{y_1, y_2, y_3\}$, а, матриця нечітких відносин має вигляд:

$$R = \begin{bmatrix} 0.8 & 0.2 & 0.3 \\ 0.7 & 0.4 & 0.5 \end{bmatrix}.$$

Допустимо в результаті діагностики системи були отримані наступні висновки: $Y = 0.8/y_1 + 0.2/y_2 + 0.3/y_3$.

Необхідно знайти передумови, що до цього призвели: $A = a_1/x_1 - a_2/x_2$.

З урахуванням конкретних даних відносини між передумовами і висновками будуть представлені в такий спосіб:

$$\begin{bmatrix} 0.8 & 0.2 & 0.3 \end{bmatrix} = \begin{bmatrix} 1 & a_2 \end{bmatrix} \bullet \begin{bmatrix} 0.8 & 0.2 & 0.3 \\ 0.7 & 0.4 & 0.5 \end{bmatrix}$$

або в транспонованому вигляді:

$$\begin{bmatrix} 0.8 \\ 0.2 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.7 \\ 0.2 & 0.4 \\ 0.3 & 0.5 \end{bmatrix} \bullet \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

При використанні max-min-композиції останнє співвідношення перетвориться до вигляду:

$$0.8 = (0.8 \cap a_1) \cup (0.7 \cap a_2),$$

$$0.2 = (0.2 \cap a_1) \cup (0.4 \cap a_2),$$

$$0.3 = (0.3 \cap a_1) \cup (0.5 \cap a_2).$$

При вирішенні даної системи помітимо, що в першому рівнянні другий член правої частини не впливає на ліву частину: $0.8 = 0.8 \cap a_1$, $a_1 \geq 0.8$.

З другого рівняння одержимо: $0.2 = 0.4 \cap a_2$, $a_2 \leq 0.2$.

Отримане рішення задовольняє третьому рівнянню. У такий спосіб: $0.8 < a_1 < 1$, $0 < a_2 < 0.2$.

При рішенні практичних задач можуть одночасно використовуватися різні правила композиції нечітких виведень. Сама схема виведень може бути багатокаскадною. В даний час загальних методів рішення подібних задач, очевидно, не існує.

A.6 Приклади роботи із функціями пакета MATLAB

Приклад 14. Створити на мові пакету MATLAB m-функцію, що реалізує колоколообразну функцію приналежності

$$\mu(x) = 1 / (1 + ((x - b)/a)^2).$$

```
function mu=bellmf(x, params)
%bellmf - колоколообразна функція приналежності;
%x - вхідний вектор;
%params(1) - коефіцієнт концентрації (>0);
%params(2) - координата максимуму.
a=params(1);
b=params(2);
mu=1./(1 + ((x - b)/a).^2);
```

Приклад 15. Використання функції fuzarith.

```
% визначення розподіляючої здатності функцій
приналежності
point_n = 101;
min_x = -20; max_x = 20;% універсальна множина є [min_x, max_x]
x = linspace(min_x, max_x, point_n)';
A = trapmf(x, [-10 -2 1 3]);% трапецієдальна нечітка множина A
B = gaussmf(x, [2 5]);% гаусівська нечітка множина B
C1 = fuzarith(x, A, B, 'sum');
subplot(2,2,1);
plot(x, A, 'y--', x, B, 'm:', x, C1, 'c');
title('нечітке додавання A+B');
C2 = fuzarith(x, A, B, 'sub');
subplot(2,2,2);
plot(x, A, 'y--', x, B, 'm:', x, C2, 'c');
title('нечітке вирахування A-B');
C3 = fuzarith(x, A, B, 'prod');
subplot(2,2,3);
plot(x, A, 'y--', x, B, 'm:', x, C3, 'c');
title('нечітке множення A*B');
C4 = fuzarith(x, A, B, 'div');
subplot(2,2,4);
plot(x, A, 'y--', x, B, 'm:', x, C4, 'c');
title('нечітке ділення A/B');
```

Зауваження. У програмах на мові MATLAB небажано використовувати кириличні літери. У наведених прикладах українськомовні фрази наведено лише для пояснення тексту програм.

Приклад 16. Використання функцій створення нейро-нечітких мереж. Апроксимувати залежність:

$$y = \frac{\sin 2x}{e^{x/5}}, x \in [0, 10], \Delta x = 0,1.$$

```
x = (0:0.1:10)'; % задаємо масив x
y = sin(2*x)./exp(x/5); % визначаємо значення елементів масиву y
trnData = [x y]; % створюємо масив навчальної вибірки
numMFs = 5; % кількість функцій приналежності
mfType = 'gbellmf'; % тип функцій приналежності
epoch_n = 20;% кількість ітерацій
% створюємо FIS-структуру типу Сугено
in_fismat = genfis1(trnData,numMFs,mfType);
% навчаємо FIS-структуру
out_fismat = anfis(trnData,in_fismat,20);
```

```

% будуємо графіки:
% цільової залежності - синій колір, суцільна лінія
% апроксимуючої залежності - зелений колір, пунктир
plot(x,y,x,evalfis(x,out_fismat));
% підписуємо легенду до графіків
% Training Data - навчальна вибірка - цільові дані
% ANFIS Output - вихід ANFIS - розрахункові дані
legend('Training Data','ANFIS Output');

```

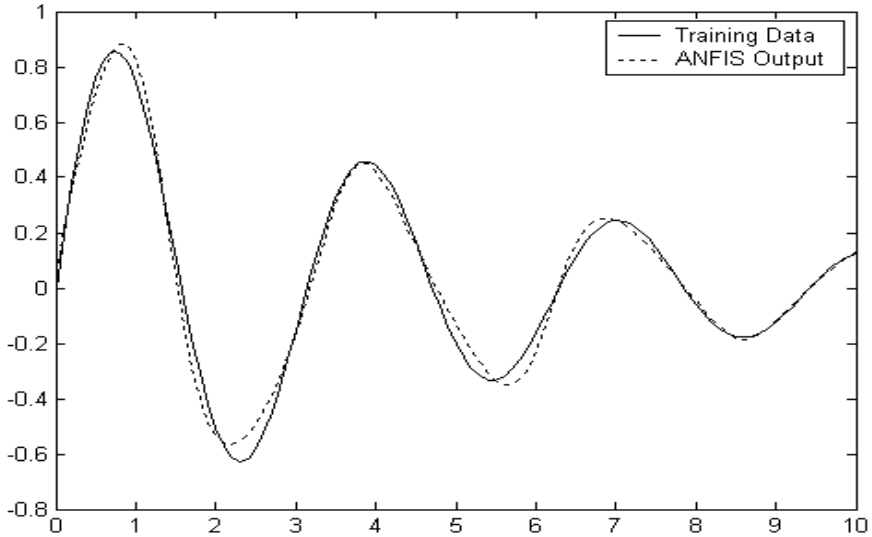


Рисунок А.2 - Графіки цільової та апроксимуючої залежностей

А.7. Побудова нечітких систем у діалоговому режимі за допомогою модуля Fuzzy Logic Toolbox

Модуль Fuzzy Logic Toolbox дозволяє будувати нечіткі системи двох типів - Мамдані та Сугено. У системах типу Мамдані база знань складається з правил виду “Якщо x_1 =низький та x_2 =середній, то y =високий”. У системах типу Сугено база знань складається з правил виду “Якщо x_1 =низький та x_2 =середній, то $y=a_0+a_1x_1+a_2x_2$ ”. Таким чином, основна відмінність між системами Мамдані та Сугено полягає в різних способах завдання значень вихідної змінної в правилах, що утворюють базу знань. У системах типу Мамдані значення вихідної змінної задаються нечіткими термами, у системах типу Сугено - як лінійна комбінація вхідних змінних.

Приклад 17. Проектування систем типу Сугено.

Розглянемо основні етапи проектування систем типу Сугено на прикладі створення системи нечіткого логічного виведення, що моделює залежність $y = x_1^2 \sin(x_2 - 1)$, $x_1 \in [-7; 3]$, $x_2 \in [-4; 4]$, $[1; 7]$. Моделювання цієї залежності будемо здійснювати за допомогою наступної бази знань:

1. Якщо x_1 =Середній, то $y=0$;
2. Якщо x_1 =Високий та x_2 =Високий, то $y=2x_1+2x_2+1$;
3. Якщо x_1 =Високий та x_2 =Низький, то $y=4x_1-x_2$;
4. Якщо x_1 =Низький та x_2 =Середній, то $y=8x_1+2x_2+8$;
5. Якщо x_1 =Низький та x_2 =Низький, то $y=50$;
6. Якщо x_1 =Низький та x_2 =Високий, то $y=50$.

Проектування системи нечіткого логічного виведення типу Сугено складається у виконанні наступної послідовності кроків.

Крок 1. Для завантаження основного fis-редактора надрукуємо слово fuzzy у командному рядку. Після цього відкриється нове графічне вікно.

Крок 2. Виберемо тип системи. Для цього в меню File вибираємо в підменю New fis...команду Sugeno.

Крок 3. Додамо другу вхідну змінну. Для цього в меню Edit вибираємо команду Add input.

Крок 4. Перейменуємо першу вхідну змінну. Для цього зробимо один щиглика лівою кнопкою миші на блоці input1, уведемо нове позначення x_1 у поле редагування імені поточної змінної і натиснемо [Enter].

Крок 5. Перейменуємо другу вхідну змінну. Для цього зробимо один щиглика лівою кнопкою миші на блоці input2, уведемо нове позначення x_2 у поле редагування імені поточної змінної і натиснемо [Enter].

Крок 6. Перейменуємо вихідну змінну. Для цього зробимо один щиглика лівою кнопкою миші на блоці output1, уведемо нове позначення y у поле редагування імені поточної змінної і натиснемо [Enter].

Крок 7. Задамо ім'я системи. Для цього в меню File вибираємо в підменю Export команду To disk і введемо ім'я файлу, наприклад, FirstSugeno.

Крок 8. Перейдемо в редактор функцій приналежності. Для цього зробимо подвійного щиглика лівою кнопкою миші на блоці x_1 .

Крок 9. Задамо діапазон зміни змінної x_1 . Для цього надрукуємо -7 3 у полі Range і натиснемо [Enter].

Крок 10. Задамо функції приналежності змінної x_1 . Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями приналежності, що встановлені за замовчуванням. Задамо найменування термів змінної x_1 . Для цього робимо один щиглик лівою кнопкою миші на графіку першої функції приналежності. Потім надрукуємо найменування терму Низький у поле Name. Потім робимо один щиглик лівою кнопкою

миші на графіку другої функції приналежності і вводимо найменування терму Середній у поле Name. Ще раз робимо один щиглик лівою кнопкою миші на графіку третьої функції приналежності і вводимо найменування терму Високий у поле Name і натиснемо [Enter].

Крок 11. Задамо функції приналежності змінної x_2 . Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями приналежності, що встановлені за замовчуванням. Для цього активізуємо змінну x_2 за допомогою щиглика лівої кнопки миші на блоці x_2 . Задамо діапазон зміни змінної x_2 . Для цього надрукуємо -4.4 1.7 у полі Range і натиснемо [Enter]. За аналогією з попереднім кроком задамо наступні найменування термів змінної x_2 : Низький, Середній, Високий.

Крок 12. Задамо лінійні залежності між входами і виходом, приведені в базі знань. Для цього активізуємо змінну y за допомогою щиглика лівої кнопки миші на блоці y . У правому верхньому куті з'явилося позначення трьох функцій приналежності, кожна з яких відповідає одній лінійній залежності між входами і виходом. У базі знань зазначені 5 різних залежностей: $y=50$; $y=4x_1-x_2$; $y=2x_1+2x_2+1$; $y=8x_1+2x_2+8$; $y=0$. Тому додамо ще дві залежності шляхом вибору команди Add Mfs...меню Edit. У діалоговому вікні, що з'явилося, в поле Number of MFs вибираємо 2 і натискаємо кнопку OK.

Крок 13. Задамо найменування і параметри лінійних залежностей. Для цього робимо один щиглик лівою кнопкою миші по найменуванню першої залежності mf1. Потім друкуємо найменування залежності, наприклад 50, у полі Name, і встановлюємо тип залежності – константа шляхом вибору опції Constant у меню Type. Після цього вводимо значення параметра – 50 у полі Params. Аналогічно для другої залежності mf2 уведемо найменування залежності, наприклад $8+8x_1+2x_2$. Потім укажемо лінійний тип залежності шляхом вибору опції Linear у меню Type і введемо параметри залежності 8 2 8 у полі Params. Для лінійної залежності порядок параметрів наступний: перший параметр – коефіцієнт при першій змінній, другий – при другій і т.д., і останній параметр – вільний член залежності. Аналогічно для третьої залежності mf3 уведемо найменування залежності, наприклад $1+2x_1+2x_2$, укажемо лінійний тип залежності і введемо параметри залежності 2 2 1. Для четвертої залежності mf4 уведемо найменування залежності, наприклад $4x_1-x_2$, укажемо лінійний тип залежності і введемо параметри залежності 4 -1 0. Для п'ятої залежності mf5 уведемо найменування залежності, наприклад 0, укажемо тип залежності – константа і введемо параметр залежності 0.

Крок 14. Перейдемо в редактор бази знань RuleEditor. Для цього виберемо в меню Edit команду Edit rules.... і введемо правила бази знань. Для введення правила необхідно вибрати відповідну комбінацію термів і залежностей і натиснути кнопку Add rule.

Вікно візуалізації нечіткого логічного виведення активізується командою View rules... меню View. У поле Input указуються значення входних змінних, для яких виконується логічне виведення. Значення вихідної змінної розраховується як середнє зважене значення результатів виведення за кожним правилом. Для виведення цього вікна поверхні «вхід-вихід», що відповідає синтезованій нечіткій системі, необхідно використовувати команду View surface... меню View.

Приклад 18. Нехай ми вирішуємо задачу визначення діагнозу пацієнта за значеннями симптомів, що характеризують його стан. Вхідними параметрами є симптоми пацієнта: температура тіла (нормальна - близько 36,6 °C, підвищена - більше 37°C), місце болю (у спині, у попереку), вид болю (сильний, слабкий). Вихідним параметром є діагноз пацієнта (хронічний апендицит, гострий апендицит, хронічний пієлонефрит, гострий пієлонефрит). Ми маємо визначені експертами правила (ми будемо вважати, що якщо апендицит або пієлонефрит не є гострими, то вони є хронічними):

ЯКЩО			ТО		
температура	місце болю	вид болю	апендицит	пієлонефрит	гострий
підвищена	у спині	сильний	немає	є	є
підвищена	у животі	сильний	є	немає	є
нормальна	у спині	слабкий	немає	є	немає
нормальна	у животі	слабкий	є	немає	немає

Необхідно побудувати систему нечіткого виведення.

1. Завантажимо пакет MATLAB та запустимо редактор FIS: fuzzy.
2. Створимо три змінні (одна вже є за замовчанням, тому додамо ще дві змінні): у головному меню редактора у підменю Edit оберемо опцію Add variables: Input. Це потрібно зробити 2 рази. Після чого створимо три вихідні змінні (подібним чином, але обираючи Add variables: Output).
3. Перемішуючи курсор миші по областях змінних можна обрати потрібну змінну та задати її назву в області Current Variable: Name. Назвемо наші змінні: temperature - температура, painlocation - місце болю, paintype - вид болю. Також задамо назви вихідних змінних: appendicite - апендицит, pielonefrit - пієлонефрит, acute - гострий.
4. Подвійним щикликом на області якої-небудь змінної можна активізувати редактор функцій приналежності. За його допомогою визначимо діапазон кожної змінної та необхідні функції приналежності.
5. У головному меню редактора FIS оберемо в підменю Edit пункт Rules. Тепер ми маємо можливість задавати наші правила.
6. У головному меню редактора FIS оберемо в підменю View пункт Rules. Перемішуючи курсором миші лінії, що регулюють значення змінних,

будемо отримувати значення вихідних змінних при відповідному наборі значень вхідних змінних.

Завантаживши файл Sample.fis із наведеним нижче текстом до редактора FIS можна переглянути описане у прикладі вирішення задачі.

Вміст файлу Sample.fis:

```
[System]
Name='Sample'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=4
NumRules=4
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'
[Input1]
Name='temperature'
Range=[35 42]
NumMFs=2
MF1='normal': 'trimf', [35.9 36.6 37]
MF2='high': 'smf', [36.8 37.19444444444444]
[Input2]
Name='painlocation'
Range=[0 1]
NumMFs=2
MF1='stiomach': 'trimf', [0 0 0.5]
MF2='spine': 'trimf', [0.5 1 1]
[Input3]
Name='paintype'
Range=[0 1]
NumMFs=2
MF1='acute': 'trimf', [0.5 1 1]
MF2='weak': 'trimf', [0 0 0.5]
[Output1]
Name='appendicite'
Range=[0 1]
NumMFs=1
MF1='is': 'trimf', [0.5 1 1]
[Output2]
Name='pielonefrit'
Range=[0 1]
```

```

NumMFs=1
MF1='is':'trimf',[0.5 1 1.4]
[Output3]
Name='acute'
Range=[0 1]
NumMFs=1
MF1='is':'trimf',[0 0 0.5]
[Output4]
Name='chronic'
Range=[0 1]
NumMFs=1
MF1='is':'trimf',[0.5 1 1.4]
[Rules]
1 1 2, 1 0 0 1 (1) : 1
1 2 2, 0 1 0 1 (1) : 1
2 1 1, 1 0 1 0 (1) : 1
2 2 1, 0 1 1 0 (1) : 1

```

А.8 Використання нейро-нечітких мереж

Приклад 19. Використання навченої нейро-нечіткої мережі.

Нехай ми маємо ємність, у яку наливається вода визначеної температури. Потрібно визначити температуру води через проміжок часу.

Входи мережі: x_1 – температура води; x_2 – час остигання води.

Вихід мережі: y – температура води після остигання.

Нехай у нас мається наступна база правил виду:

Якщо вода x_1 та остигає x_2 , то температура води буде y .

Представимо її у вигляді таблиці:

НОМЕР ПРАВИЛА	ЯКЩО (входи)		ТО (вихід)	ВАГА ПРАВИЛА
	x_1	x_2		
11	холодна	недовго	прийнятна	1
12	холодна	довго		1
13	тепла	довго		0,8
14	гаряча	довго	нормальна	0,6
21	тепла	недовго		0,5
22	гаряча	довго		0,8
31	гаряча	недовго	висока	0,9

Тут холодна, тепла, гаряча, довго, недовго – нечіткі терми (нечіткі множини):

$$\mu_{\text{хол}}(x) = \begin{cases} 1, & x \leq 15^\circ\text{C}, \\ -\frac{x}{15} + 2, & 15 < x < 30^\circ\text{C}, \\ 0, & x \geq 30^\circ\text{C}, \end{cases} \quad \mu_{\text{тепл}}(x) = \begin{cases} \frac{x}{15} - 1.25, & x \leq 35^\circ\text{C}, \\ -\frac{x}{15} + \frac{10}{3}, & x > 35^\circ\text{C}, \end{cases}$$

$$\mu_{\text{гар}}(x) = \begin{cases} \frac{x}{15} - \frac{8}{3}, & x \leq 40^\circ\text{C}, \\ 1, & x > 40^\circ\text{C}, \end{cases}$$

$$\mu_{\text{нед}}(x) = \begin{cases} 1, & x \leq 10 \text{ хв.}, \\ -\frac{x}{10} + 2, & x > 10 \text{ хв.}, \end{cases} \quad \mu_{\text{дов}}(x) = \begin{cases} \frac{x}{10} - 1, & x \leq 20 \text{ хв.}, \\ 1, & x > 20 \text{ хв.}, \end{cases}$$

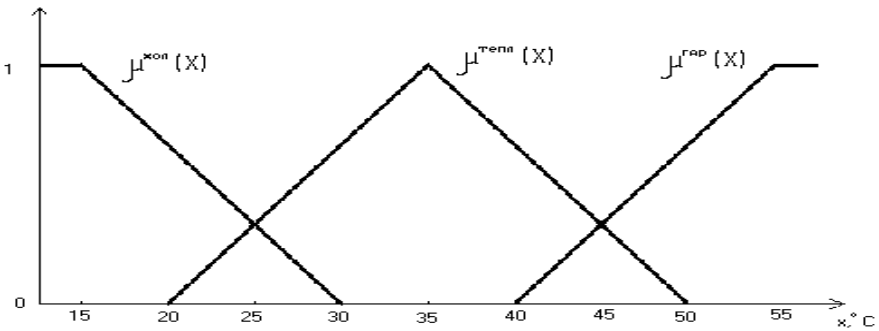


Рисунок А.1 - Функції приналежності: $\mu^{\text{хол}}(x)$, $\mu^{\text{тепл}}(x)$, $\mu^{\text{гар}}(x)$

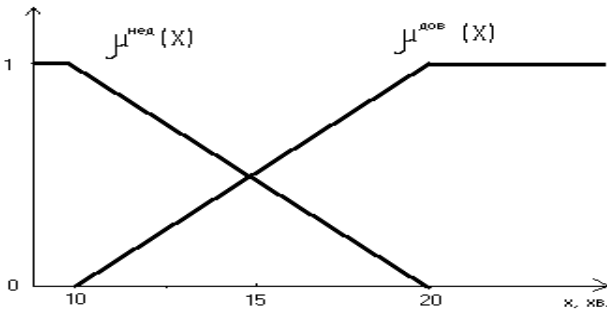


Рисунок.А.2 - Функції приналежності $\mu^{\text{дол}}(x)$ та $\mu^{\text{нед}}(x)$

Розбиття на класи для y виглядає таким чином:

прийнятна: [10; 25]

нормальна: [25; 45]

висока: [45; 60]

Таким чином: $\underline{y} = 10, y_1 = 25, y_2 = 45, \bar{y} = 60$.

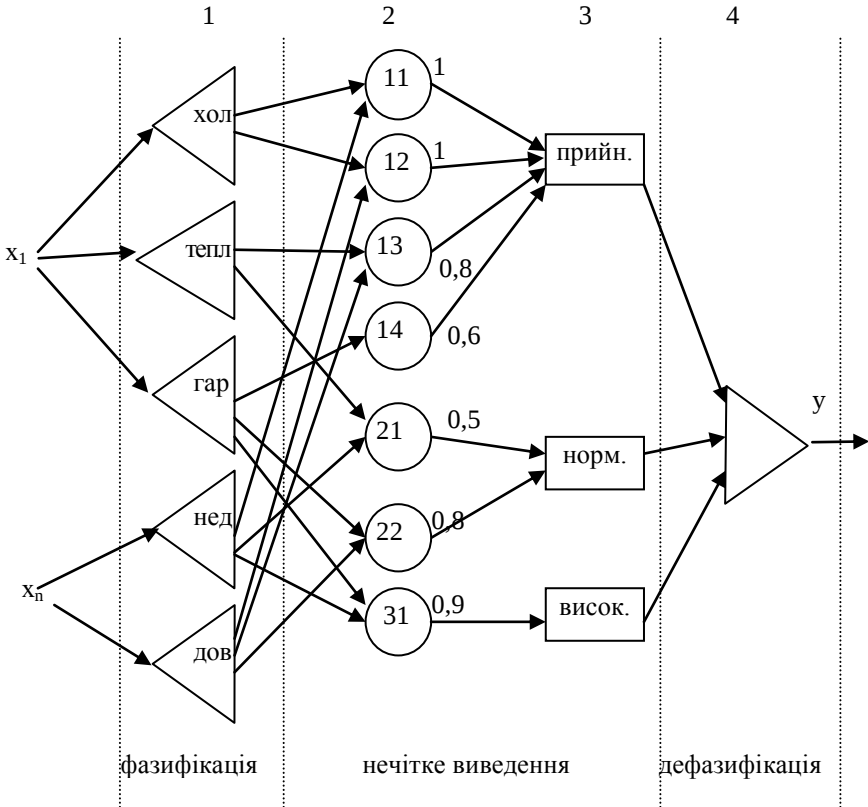


Рисунок А.3 - Нечітка нейронна мережа

Шар 1. Визначаються функції приналежності вхідних змінних термам. На виході $\mu^T(x_i), i = 1, 2$.

Шар 2. Вибирається мінімальний із входів нейрона. На виході $\min(\mu^T(x_i)), j = 1, 2, 3$ – номер класу y .

Шар 3. Вибирається максимальний із входів нейрона (вхід є добутком виходу нейрона попереднього шару і ваги правила). На виході $\max(w_{jp}, \min(\mu^T(x_i)))$, $p = 1, \dots, 7$ – номер правила.

Шар 4. Визначаємо вихід:

$$y = \frac{y \cdot \mu^{d1}(y) + y_1 \cdot \mu^{d2}(y) + y_2 \cdot \mu^{d3}(y)}{\mu^{d1}(y) + \mu^{d2}(y) + \mu^{d3}(y)}.$$

Нехай $x_1 = 22$ і $x_2 = 12$, тоді:

визначення приналежностей термам (фазифікація):

$$\mu_{\text{хол}}(x_1) = 0,53; \mu_{\text{тепл}}(x_1) = 0,22; \mu_{\text{гар}}(x_1) = 0; \mu_{\text{нед}}(x_2) = 0,8; \mu_{\text{дов}}(x_2) = 0,2.$$

результати виконання нечітких правил:

$$11: \min(0,53, 0,8) = 0,53;$$

$$12: \min(0,53, 0,2) = 0,2;$$

$$13: \min(0,22, 0,2) = 0,2;$$

$$14: \min(0, 0,2) = 0;$$

$$21: \min(0,22, 0,8) = 0,22;$$

$$22: \min(0, 0,22) = 0;$$

$$31: \min(0, 0,8) = 0.$$

результати об'єднання правил у класи:

$$d_1: \max(1 \cdot 0,53, 1 \cdot 0,2, 0,8 \cdot 0,2, 0,6 \cdot 0) = 0,53$$

$$d_2: \max(0,5 \cdot 0,22, 0,8 \cdot 0) = 0,11; \quad d_3: \max(0,9 \cdot 0) = 0.$$

Дефазифікація:

$$y = (10 \cdot 0,53 + 25 \cdot 0,11) / (0,53 + 0,11) = 12,6.$$

Одержуємо відповідь: вода 22°C після остигання протягом 12 хвилин буде мати температуру 12,6°C.

Приклад 20. Нехай ми маємо спостереження за погодою, представлені наборами значень змінних x_1 - температура повітря, °C; x_2 - вміст вологи у повітрі, %; y - кількість опадів, мм/см². Ми знаємо, що параметр y залежить від параметрів x_1 та x_2 , але ми не знаємо як. Треба побудувати нейро-нечітку апроксимацію залежності $y(x_1, x_2)$ за даними спостережень, представленими у вигляді таблиці (навчальна вибірка). Для перевірки апроксимаційних властивостей нейро-нечіткої моделі будемо використовувати тестову вибірку, представлену у таблиці.

навчаюча вибірка		
x ₁	x ₂	y
25	76	2.3
27	79	2.2
23.5	91	1.8
15	89	1.2
14.5	79	1.0
17	61	1.4
20	96	2.0
23	89	2.0
26.5	57	2.6
29	59	2.7

тестова вибірка		
x ₁	x ₂	y
15	0.86	0.9
25	0.47	2.5
26	0.57	2.6
15	0.98	0.9
13.5	0.68	0.9
16.5	0.93	1.6
19.5	0.67	1.9
22	0.58	2.1
25	0.82	2.0
28	0.86	2.8

У середовищі MATLAB створимо змінну навчаючої вибірки *p* та змінну тестової вибірки *t*:

```
p=[25 76 2.3; 27 79 2.2; 23.5 91 1.8; 15 89 1.2; 14.5 79
1.0; 17 61 1.4; 20 96 2.0; 23 89 2.0; 26.5 57 2.6; 29 59
2.7];
```

```
t=[15 0.86 0.9; 25 0.47 2.5; 26 0.57 2.6; 15 0.98 0.9;
13.5 0.68 0.9; 16.5 0.93 1.6; 19.5 0.67 1.9; 22 0.58 2.1; 25
0.82 2.0; 28 0.86 2.8];
```

Запустимо редактор ANFIS, для чого введемо: `anfisedit`

В області Load data задамо змінні тестової вибірки (оберемо в списку Type: Training та в списку from: Worksp., після чого натиснемо кнопку Load data .. та введемо у поле input variable name: *p*) та навчаючої вибірки (оберемо в списку Type: Testing та в списку from: Worksp., після чого натиснемо кнопку Load data .. та введемо у поле input variable name: *t*).

В області Generate Fis оберемо: Sub. clustering та натиснемо кнопку Generate Fis. В області Train FIS оберемо метод оптимізації Optim. method: hybrid. Задамо прийнятний рівень помилки Error tolerance: 0.000001. Задамо кількість ітерацій Epochs: 300. Натиснемо кнопку Train now. У графічній області Training Error будемо спостерігати, як змінюється помилка нейро-нечіткої мережі протягом навчання. Натиснувши кнопку Structure отримаємо зображення структури згенерованої нейро-нечіткої мережі.

Після цього отриману FIS структуру можна зберегти на диску або використовувати для прогнозування *y*, наприклад із використанням тестової вибірки *t*. Для тестування оберемо, для якої вибірки здійснюється тестування (Training - навчальна, Testing - тестова), та натиснемо кнопку Test now. Внизу форми буде вказано середню помилку тестування (Average Error), а на графіку буде зображено синіми точками - цільові значення виходу, а червоними зірками - розрахункові значення виходу нейро-нечіткої мережі.

Додаток Б

Нечітка логіка та нейро-нечіткі мережі у пакеті MATLAB

Операції з нечіткою логікою у пакеті MATLAB дозволяє виконувати модуль *Fuzzy Logic Toolbox*. Він дозволяє створювати системи нечіткого логічного виведення і нечіткої класифікації в рамках середовища MatLab, з можливістю їхнього інтегрування в Simulink.

Fuzzy Logic Toolbox містить наступні категорії програмних інструментів: функції; інтерактивні модулі з графічним користувацьким інтерфейсом (з GUI); блоки для пакета Simulink; демонстраційні приклади.

Б.1 FIS - структура

Базовим поняттям Fuzzy Logic Toolbox є *FIS-структура* - система нечіткого виведення (Fuzzy Inference System). FIS-структура містить усі необхідні дані для реалізації функціонального відображення “входи-виходи” на основі нечіткого логічного виведення відповідно до схеми, приведеної на рис. Б.1.

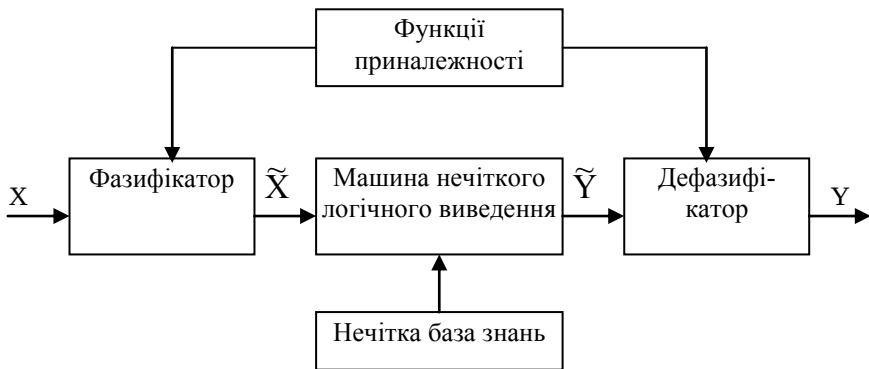


Рисунок Б.1 - Нечітке логічне виведення

Позначення: X - вхідний чіткий вектор; $\tilde{X}$ - вектор нечітких множин, що відповідає вхідному вектору X ; $\tilde{Y}$ - результат логічного виведення у виді вектора нечітких множин; Y - вихідний чіткий вектор.

Система нечіткого логічного виведення представляється в робочій області MatLab у вигляді структури даних, зображеної на рис. Б.2.

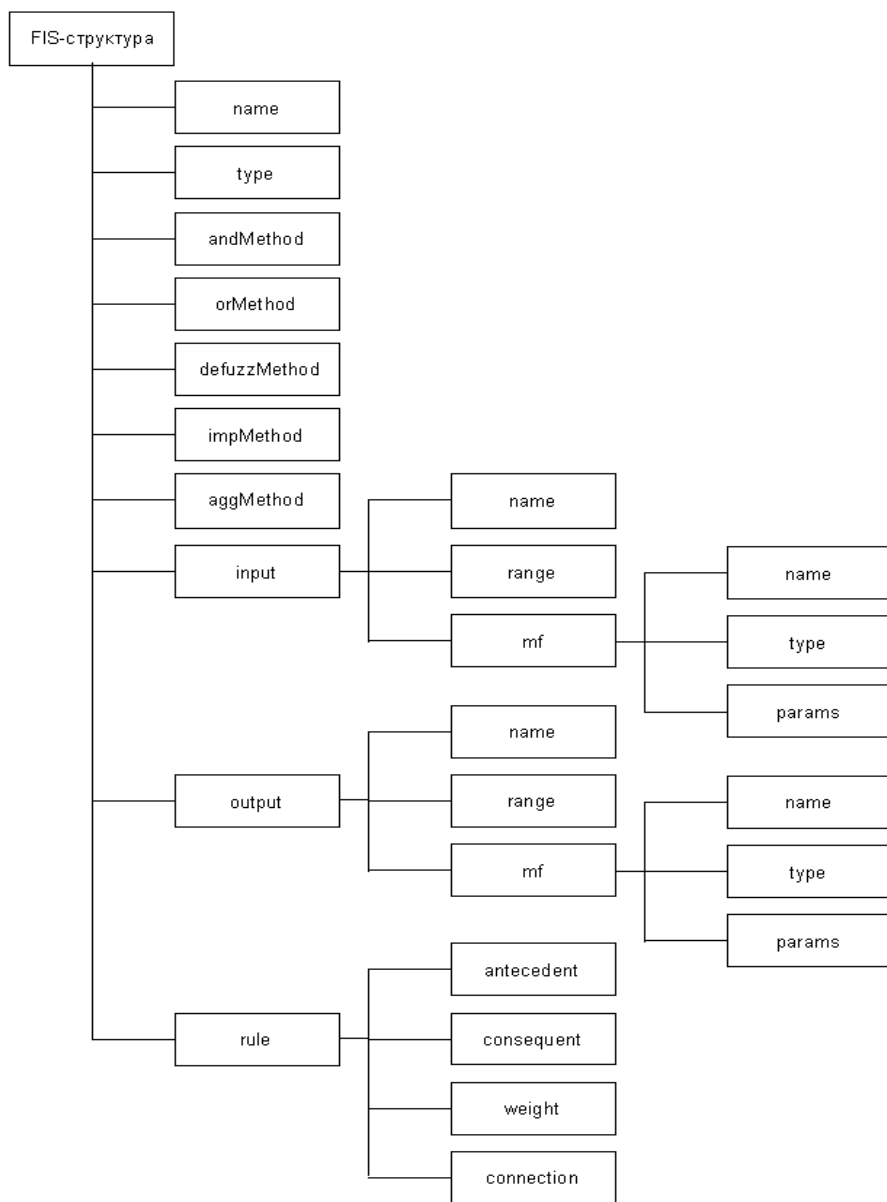


Рисунок Б.2 - FIS-структура

Існує два *способи завантаження FIS* у робочу область: зчитування з диска за допомогою функції `readfis`; передача з основного `fis`-редактора шляхом вибору в меню `File` підміну `Export` і команди `To workspace`.

Поля структури дані системи нечіткого логічного виведення призначені для збереження наступної інформації:

- `name` - найменування системи нечіткого логічного виведення;
- `type` - тип системи (припустимі значення `'Mamdani'` та `'Sugeno'`);
- `andMethod` - реалізація логічної операції "ТА" (запрограмовані реалізації: `'min'` – мінімум і `'prod'` – множення);
- `orMethod` - реалізація логічної операції "АБО" (запрограмовані реалізації: `'max'` – максимум і `'probor'` – імовірнісне "АБО");
- `defuzzMethod` - метод дефазифікації (запрограмовані методи для систем типу Мамдані: `'centroid'` – центр ваги; `'bisector'` – медіана; `'lom'` – найбільший з максимумів; `'som'` – найменший з максимумів; `'mom'` – середнє з максимумів; запрограмовані методи для систем типу Сугено: `'wtaver'` – зважене середнє і `'wtsum'` – зважена сума);
- `impMethod` - реалізація операції імплікації (запрограмовані реалізації: `'min'` – мінімум і `'prod'` – множення);
- `aggMethod` - реалізація операції об'єднання функцій приналежності вихідної змінної (запрограмовані реалізації: `'max'` – максимум; `'sum'` – сума і `'probor'` – імовірнісне "АБО");
- `input` - масив вхідних змінних;
- `input.name` - найменування вхідної змінної;
- `input.range` - діапазон зміни вхідної змінної;
- `input.mf` - масив функцій приналежності вхідної змінної;
- `input.mf.name` - найменування функції приналежності вхідної змінної;
- `input.mf.type` - модель функції приналежності вхідної змінної (запрограмовані моделі: `dsigmf` - функція приналежності у виді різниці між двома сигмоїдними функціями; `gauss2mf` - двостороння гаусівська функція приналежності; `gaussmf` - гаусівська функція приналежності; `gbellmf` - узагальнена колоколообразна функція приналежності; `rimf` - пі-подібна функція приналежності; `psigmf` - добуток двох сигмоїдних функцій приналежності; `sigmf` - сигмоїдна функція приналежності; `smf` - s-подібна функція приналежності; `trapmf` - трапецієподібна функція приналежності; `trimf` - трикутна функція приналежності; `zmf` - z-подібна функція приналежності);
- `input.mf.params` - масив параметрів функції приналежності вхідної змінної;
- `output` - масив вихідних змінних;

- output.name - найменування вихідної змінної;
- output.range - діапазон зміни вихідної змінної;
- output.mf - масив функцій приналежності вихідної змінної;
- output.mf.name - найменування функції приналежності вихідної змінної;
- output.mf.type - модель функції приналежності вихідної змінної (запрограмовані моделі для системи типу Мамдані: dsigmf - функція приналежності у виді різниці між двома сигмоїдними функціями; gauss2mf - двостороння гаусівська функція приналежності; gaussmf - гаусівська функція приналежності; gbellmf - узагальнена колоколообразна функція приналежності; pimf - пі-подібна функція приналежності; psigmf - добуток двох сигмоїдних функцій приналежності; запрограмовані моделі для системи типу Сугено: constatnt – константа (функція приналежності у виді сінглтона); linear – лінійна комбінація вхідних змінних);
- output.mf.params - масив параметрів функції приналежності вихідної змінної;
- rule - масив правил нечіткої бази знань;
- rule.antecedent - посилки правила (вказуються порядкові номери термів у порядку запису вхідних змінних. Число 0 указує на те, що значення відповідної вхідний змінної не впливає на істинність правила);
- rule.consequent - висновок правила (вказуються порядкові номери термів у порядку запису вихідних змінних. Число 0 указує на те, що правило не поширюється на відповідну вихідну змінну);
- rule.weight - вага правила. Задається числом з діапазону [0, 1];
- rule.connection - логічне зв'язування змінних усередині правила: 1 – логічне "ТА"; 2 – логічне "АБО".

Для доступу до властивостей системи нечіткого логічного виведення досить вказати ім'я відповідного поля. Наприклад, команда FIS_NAME.rule(1).weight=0.5 встановлює вагу першого правила в 0.5, команда length(FIS_NAME.rule) визначає кількість правил у базі знань, а команда FIS_NAME.input(1).mf(1).name='низький' перейменовує перший терм першої вхідної змінної в "низький".

Б.2 FIS-редактор

FIS-редактор (рис. Б.3) призначений для створення, збереження, завантаження і виведення у друк систем нечіткого логічного виведення, а також для редагування наступних властивостей: тип системи; найменування системи; кількість вхідних і вихідних змінних; найменування вхідних і вихідних змінних; параметри нечіткого логічного виведення.

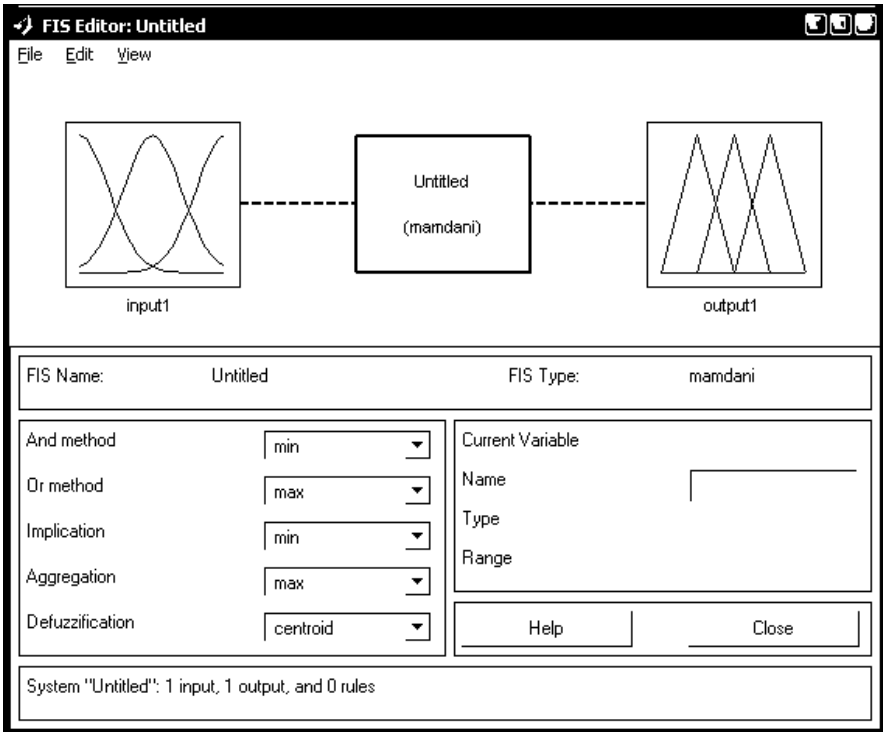


Рисунок Б.3 - FIS-редактор

Завантаження FIS-редактора відбувається за допомогою команди fuzzy. У результаті з'являється інтерактивне графічне вікно. У нижній частині графічного вікна FIS-редактори розташовані кнопки Help і Close, що дозволяють викликати вікно довідки і закрити редактор, відповідно.

FIS-редактор містить 8 меню. Це три загальносистемних меню - File, Edit, View, і п'ять меню для вибору параметрів нечіткого логічного виведення – And Method, Or Method, Implication, Aggregation і Defuzzification.

Меню File - це загальне меню для всіх GUI-модулів використовуваних із системами нечіткого логічного виведення.

За допомогою команди New FIS... користувач має можливість створити нову систему нечіткого логічного виведення. При виборі цієї команди з'являються дві альтернативи: Mamdani і Sugeno, що визначають тип створюваної системи. Створити систему типу Mamdani можна також натисканням Ctrl+N.

За допомогою команди Import користувач має можливість завантажити раніше створену систему нечіткого логічного виведення. При виборі цієї команди з'являться дві альтернативи From Workspace... і From disk, що дозволяють завантажити систему нечіткого логічного виведення з робочої області MatLab і з диска, відповідно. При виборі команди From Workspace... з'явиться діалогове вікно, у якому необхідно вказати ідентифікатор системи нечіткого логічного виведення, що знаходиться в робочій області MatLab. При виборі команди From disk з'явиться діалогове вікно, у якому необхідно вказати ім'я файлу системи нечіткого логічного виведення. Файли систем нечіткого логічного виведення мають розширення .fis . Завантажити систему нечіткого логічного виведення з диска можна також натисканням Ctrl+N чи командою fuzzy FIS_name, де FIS_name – ім'я файлу системи нечіткого логічного виведення.

При виборі команди Export з'являться дві альтернативи To Workspace... і To disk, що дозволяють скопіювати систему нечіткого логічного виведення в робочу область MatLab і на диск, відповідно. При виборі команди To Workspace... з'явиться діалогове вікно, у якому необхідно вказати ідентифікатор системи нечіткого логічного виведення, під яким вона буде збережена в робочій області MatLab. При виборі команди To disk з'явиться діалогове вікно, у якому необхідно вказати ім'я файлу системи нечіткого логічного виведення. Скопіювати систему нечіткого логічного виведення в робочу область і на диск можна також натисканням Ctrl+T і Ctrl+S, відповідно.

Команда Print дозволяє вивести на принтер копію графічного вікна. Печатка можлива також по натисканню Ctrl+P.

Команда Close закриває графічне вікно. Закриття графічного вікна відбувається по натисканню Ctrl+W чи однократного щиглика лівої кнопки миші по кнопці Close.

Меню Edit:

Команда Undo скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда Add Variable... дозволяє додати в систему нечіткого логічного виведення ще одну змінну. При виборі цієї команди з'являться дві альтернативи Input і Output, що дозволяють додати вхідну і вихідну змінну, відповідно.

Команда Remove Selected Variable видаляє поточну змінну із системи. Ознакою поточної змінної є червона окантовка її прямокутника. Призначення поточної змінної відбувається за допомогою однократного щиглика лівої кнопки миші по її прямокутнику. Видалити поточну змінну можна також за допомогою натискання Ctrl+X.

Команда *Membership Function...* відкриває редактор функцій приналежностей. Ця команда може бути також виконана натисканням **Ctrl+2**.

Команда *Rules...* відкриває редактор бази знань. Ця команда може бути також виконана натисканням **Ctrl+3**.

Меню View - це загальне меню для всіх GUI-модулів, використовуваних із системами нечіткого логічного виведення. Дане меню дозволяє відкрити вікно візуалізації нечіткого логічного виведення (команда *Rules* або натискання клавіш **Ctrl+5**) і вікно виведення поверхні "вхід-вихід", що відповідає системі нечіткого логічного виведення (команда *Surface* або натискання клавіш **Ctrl+6**).

Меню And Method - це меню дозволяє установити наступні реалізації логічної операції "ТА": *min* – мінімум; *prod* – множення.

Користувач також має можливість установити власну реалізацію операції "ТА". Для цього необхідно вибрати команду *Custom...* і в графічному вікні, що з'явилося, надрукувати ім'я функції, що реалізує цю операцію.

Меню Or Method - це меню дозволяє установити наступні реалізації логічної операції "АБО": *max* – множення; *probor* - імовірнісне "АБО".

Користувач також має можливість установити власну реалізацію операції "АБО". Для цього необхідно вибрати команду *Custom...* і в графічному вікні, що з'явилося, надрукувати ім'я функції, що реалізує цю операцію.

Меню Implication - це меню дозволяє установити наступні реалізації імплікації: *min* – мінімум; *prod* – множення.

Користувач також має можливість установити власну реалізацію імплікації. Для цього необхідно вибрати команду *Custom...* і в графічному вікні, що з'явилося, надрукувати ім'я функції, що реалізує цю операцію.

Меню Aggregation - це меню дозволяє установити наступні реалізації операції об'єднання функцій приналежності вихідної змінної: *max* – максимум; *sum* – сума; *probor* - імовірнісне "АБО".

Користувач також має можливість установити власну реалізацію цієї операції. Для цього необхідно вибрати команду *Custom...* і в графічному вікні, що з'явилося, надрукувати ім'я функції, що реалізує цю операцію.

Меню Defuzzification - це меню дозволяє вибрати метод дефазифікації. Для систем типу Мамдані запрограмовані наступні методи: *centroid* – центр ваги; *bisector* – медіана; *lom* – найбільший з максимумів; *som* – найменший з максимумів; *mom* – середнє з максимумів. Для систем типу Сугено запрограмовані наступні методи: *wtaver* – зважене середнє; *wtsum* – зважена сума.

Користувач також має можливість установити власний метод дефазифікації. Для цього необхідно вибрати команду *Custom...* і в графічному вікні, що з'явилося, надрукувати ім'я функції, що реалізує цю операцію.

Б.3 Редактор функцій приналежності

Редактор функцій приналежності (Membership Function Editor) призначений для завдання наступної інформації про терми-множини вхідних і вихідних змінних: кількість термів; найменування термів; тип і параметри функцій приналежності, що необхідні для представлення лінгвістичних термів у вигляді нечітких множин.

Редактор функцій приналежності може бути викликаний з будь-якого GUI-модуля, використовуюваного із системами нечіткого логічного виведення, командою Membership Functions... меню Edit або натисканням клавіш Ctrl+2.

У FIS-редакторі відкрити редактор функцій приналежності можна також подвійним щикликом лівою кнопкою миші по полю вхідної або вихідної змінних. Загальний вид редактора функцій приналежності з указівкою функціонального призначення основних полів графічного вікна приведений на рис. Б.4.

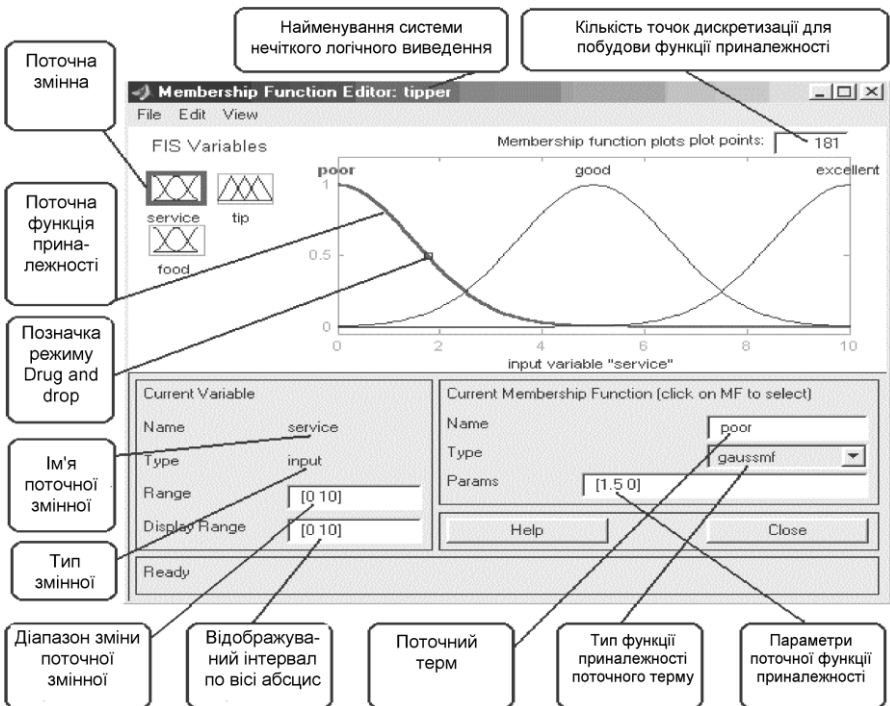


Рисунок Б.4 - Редактор функцій приналежності

У нижній частині графічного вікна розташовані кнопки Help і Close, що дозволяють викликати вікно довідки і закрити редактор, відповідно.

Редактор функцій приналежності містить чотири меню - File, Edit, View, Type і чотири вікна введення інформації – Range, Display Range, Name і Params. Ці чотири вікна призначені для завдання діапазону зміни поточної змінної, діапазону виведення функцій приналежності, найменування поточного лінгвістичного терму і параметрів його функції приналежності, відповідно.

Параметри функції приналежності можна підбирати й у графічному режимі, шляхом зміни форми функції приналежності за допомогою технології “Drug and drop”. Для цього необхідно позиціонувати курсор миші на знаку режиму “Drug and drop”, натиснути на ліву кнопку миші і не відпускаючи її змінювати форму функції приналежності. Параметри функції приналежності будуть перераховуватися автоматично.

Меню Edit:

Команда Undo скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда Add MFs...дозволяє додати терми в терм-множину, використовувана для лінгвістичної оцінки поточної змінної. При виборі цієї команди з'явиться діалогове вікно, у якому необхідно вибрати тип функції приналежності і кількість термів. Значення параметрів функцій приналежності будуть встановлені автоматично таким чином, щоб рівномірно покрити область визначення змінної, заданої у вікні Range. При зміні області визначення у вікні Range параметри функцій приналежності будуть промаштабовані.

Команда Add Custom MF...дозволяє додати одних лінгвістичний терм, функція приналежності якого відрізняється від убудованих. Після вибору цієї команди з'явиться графічне вікно, у якому необхідно надрукувати лінгвістичний терм (поле MF name), ім'я функції приналежності (поле M-File function name) і параметри функції приналежності (поле Parameter list).

Команда Remove Selected MF видаляє поточний терм із терм-множини поточної змінної. Ознакою поточної змінної є червона окантовка її прямокутника. Ознакою поточного терму є червоний колір його функції приналежності. Для вибору поточного терму необхідно провести позиціонування курсору миші на графіку функції приналежності і зробити щиклик лівою кнопкою миші.

Команда Remove All MFs видаляє всі терми з терм-множини поточної змінної.

Команда FIS Properties...відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+I.

Команда Rules...відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Меню *Type* дозволяє установити тип функцій приналежності термів, використовуваних для лінгвістичної оцінки поточної змінної.

Б.4 Редактор бази знань

Редактор бази знань (Rule Editor) призначений для формування і модифікації нечітких правил. Редактор бази знань може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виведення, командою Rules...меню Edit або натисканням клавіш Ctrl+3. У FIS-редакторі відкрити редактор бази знань можна також подвійним щикликом лівою кнопкою миші по прямокутнику з назвою системи нечіткого логічного виведення, розташованого в центрі графічного вікна.

Загальний вид редактора бази знань із указівкою функціонального призначення основних полів графічного вікна приведений на рис. Б.5.

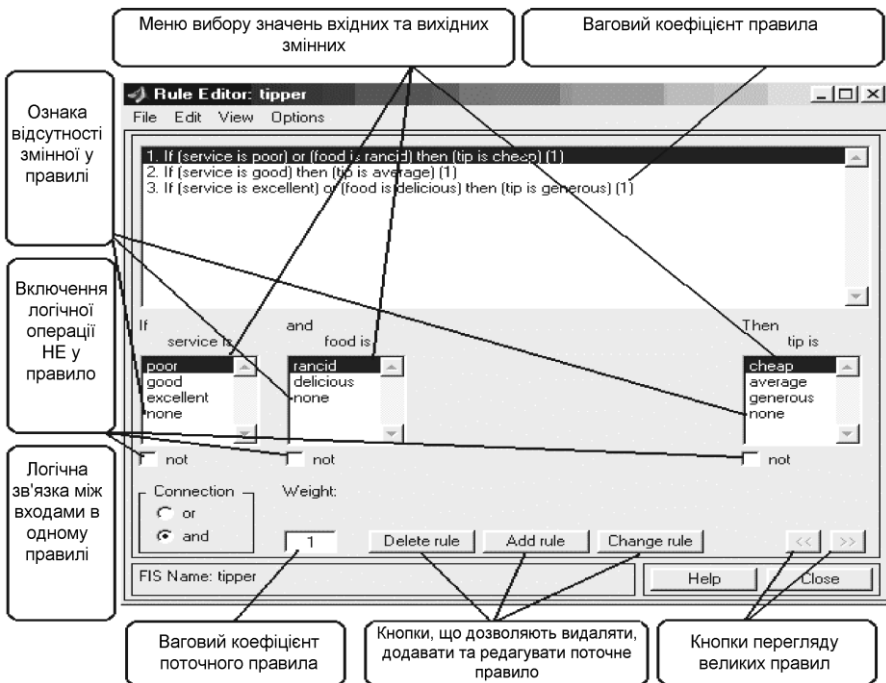


Рисунок Б.5 - Редактор бази знань

У нижній частині графічного вікна розташовані кнопки Help і Close, що дозволяють викликати вікно довідки і закрити редактор, відповідно.

Редактор функцій приналежності містить чотири системних меню File, Edit, View, Options, меню вибору термів вхідних і вихідних змінних, поля установки логічних операцій ТА, АБО, НЕ і ваг правил, а також кнопки редагування і перегляду правил.

Для введення нового правила в базу знань необхідно за допомогою миші вибрати відповідну комбінацію лінгвістичних термів вхідних і вихідних змінних, установити тип логічного зв'язування (ТА або АБО) між змінними усередині правила, установити чи наявність відсутність логічної операції НЕ для кожної лінгвістичної змінної, увести значення вагового коефіцієнта правила і натиснути кнопку Add Rule. За замовчуванням установлені наступні параметри: логічне зв'язування змінних усередині правила – ТА; логічна операція НЕ – відсутня; значення вагового коефіцієнта правила – 1.

Можливі випадки, коли істинність правила не змінюється при довільній значенні деякої вхідної змінної, тобто ця змінна не впливає на результат нечіткого логічного виведення в даній області факторного простору. Тоді як лінгвістичне значення цієї змінної необхідно установити none.

Для видалення правила з бази знань необхідно зробити однократний щиклик лівою кнопкою миші на цьому правилі та натиснути кнопку Delete Rule.

Для модифікації правила необхідно зробити однократний щиклик лівою кнопкою миші на цьому правилі, потім установити необхідні параметри правила і натиснути кнопку Edit Rule.

Меню Edit:

Команда Undo скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда FIS Properties...відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда Membership Function...відкриває редактор функцій приналежностей. Ця команда може бути також виконана натисканням Ctrl+2.

Меню Options дозволяє установити мову і формат правил бази знань. При виборі команди Language з'явиться список мов English (Англійська), Deutsch (Німецька), Francais (Французька), з якого необхідно вибрати одну.

При виборі команди Format з'явиться список можливих форматів правил бази знань: Verbose - лінгвістичний; Symbolic – логічний; Indexed – індексований.

Б.5 Візуалізація нечіткого логічного виведення

Візуалізація нечіткого логічного виведення здійснюється за допомогою GUI-модуля *Rule Viewer*. Цей модуль дозволяє проілюструвати хід логічного виведення за кожним правилом, одержання результуючої

нечіткої множини і виконання процедури дефазифікації. Rule Viewer може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виведення, командою View rules ... меню View чи натисканням клавіш Ctrl+4. Вид Rule Viewer для системи логічного виведення tipper із указівкою функціонального призначення основних полів графічного вікна приведений на рис. Б.6.

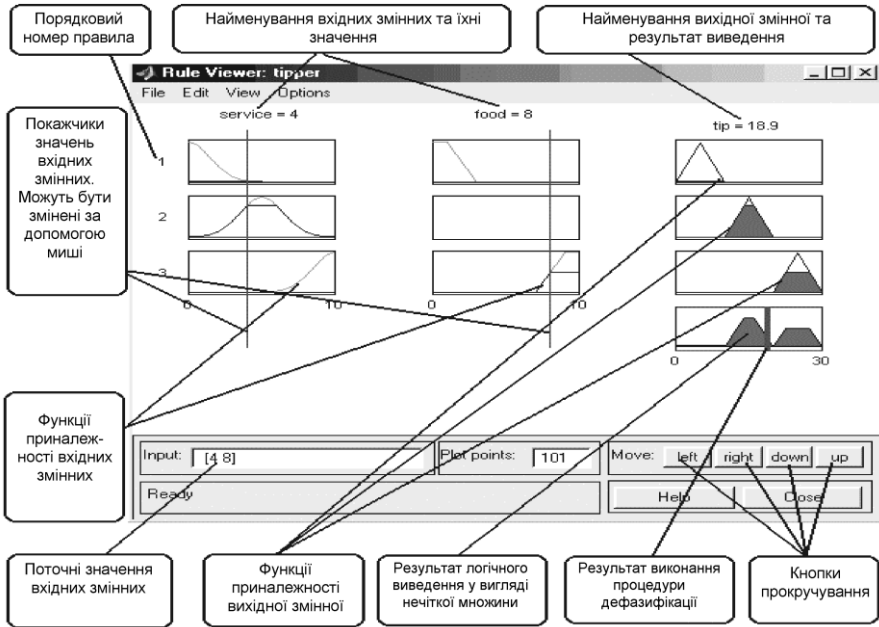


Рисунок Б.6 – Візуалізація логічного виведення для системи tipper за допомогою Rule Viewer

Rule Viewer містить чотири меню - File, Edit, View, Options, два поля уведення інформації – Input і Plot points та кнопки прокручування зображення вліво - вправо (left-right), униз (up-down). У нижній частині графічного вікна розташовані також кнопки Help і Close, що дозволяють викликати вікно довідки і закрити редактор, відповідно. Кожне правило бази знань представляється у виді послідовності горизонтально розташованих прямокутників. При цьому перші два прямокутники відображають функції приналежностей термів посилки правила (Якщо-частина правила), а останній третій прямокутник відповідає функції приналежності терму-наслідку вихідної змінної (То-частина правила).

Порожній прямокутник у візуалізації другого правила означає, що в цьому правилі посилка за змінною *food* відсутня (*food is none*). Жовте заливання графіків функцій приналежностей вхідних змінних указує наскільки значення входів, відповідають термам даного правила. Для виведення правила у форматі Rule Editor необхідно зробити однократний щиклик лівої кнопки миші по номері відповідного правила. У цьому випадку зазначене правило буде виведено в нижній частині графічного вікна.

Блакитне заливання графіка функції приналежності вихідної змінної являє собою результат логічного виведення у вигляді нечіткої множини за даним правилом. Результуючу нечітку множину, що відповідає логічному виведенню за всіма правилами, показано в нижньому прямокутнику останнього стовпця графічного вікна. У цьому ж прямокутнику червона вертикальна лінія відповідає чіткому значенню логічного виведення, отриманого в результаті дефазифікації.

Уведення значень вхідних змінних може здійснюватися двома способами: шляхом введення чисельних значень у поле Input; за допомогою миші, шляхом переміщення ліній-показчиків червоного кольору.

В останньому випадку необхідно позиціонувати курсор миші на червоній вертикальній лінії, натиснути на ліву кнопку миші і не відпускаючи неї перемістити показчик на потрібну позицію. Нове чисельне значення відповідної вхідної змінної буде перелічено автоматично і виведено у вікно Input.

У поле Plot points задається кількість крапок дискретизації для побудови графіків функцій приналежності. Значення за замовчуванням – 101.

Меню Edit:

Команда FIS Properties...відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда Membership Functions...відкриває редактор функцій приналежностей. Ця команда може бути також виконана натисканням Ctrl+2.

Команда Rules...відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Меню Options містить тільки одну команду Format, що дозволяє установити один з наступних форматів виведення обраного правило в нижній частині графічного вікна: Verbose – лінгвістичний; Symbolic – логічний; Indexed – індексований.

Б.6 ANFIS-редактор

ANFIS є аббревіатурою Adaptive Neuro-Fuzzy Inference System – адаптивна нейро-нечітка система. ANFIS-редактор дозволяє автоматично синтезувати з експериментальних даних нейро-нечіткої мережі. Нейро-нечітку мережу можна розглядати як один з різновидів систем нечіткого логічного

виведення типу Сугено. При цьому функції приналежності синтезованих систем налагоджено (навчено) так, щоб мінімізувати відхилення між результатами нечіткого моделювання й експериментальних даних.

Завантаження ANFIS-редактора здійснюється за командою `anfisedit`. У результаті виконання цієї команди з'явиться графічне вікно, зображене на рис. Б.7. На цьому ж рисунку зазначено функціональні області ANFIS-редактора, опис яких приведено нижче.

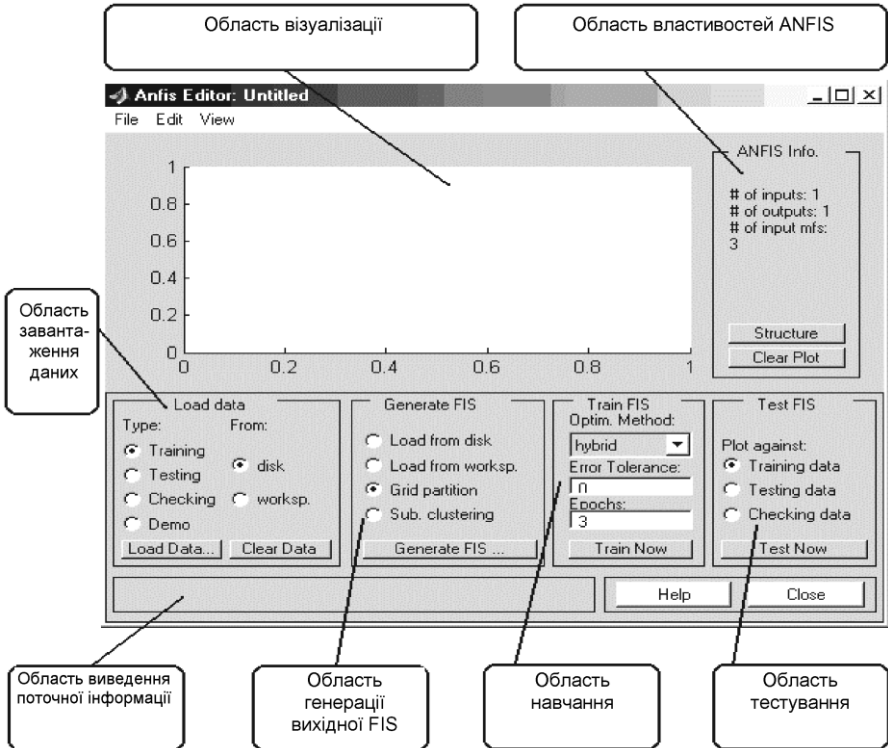


Рисунок Б.7 - Основне вікно ANFIS-редактора

ANFIS-редактор містить 3 верхніх меню - File, Edit і View, область візуалізації, область властивостей ANFIS, область завантаження даних, область генерування вихідної системи нечіткого логічного виведення, область навчання, область тестування, область виведення поточної інформації, а також кнопки Help і Close, що дозволяють викликати вікно довідки і закрити ANFIS-редактор, відповідно.

Меню Edit:

Команда Undo скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда FIS Properties...відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда Membership Functions...відкриває редактор функцій приналежності. Ця команда може бути також виконана натисканням Ctrl+2.

Команда Rules...відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Команда Anfis...відкриває ANFIS-редактор. Ця команда може бути також виконана натисканням Ctrl+3. Помітимо, що дана команда, запущена з ANFIS-редактора не приводить до виконання яких-небудь дій, так цей редактор уже відкритий. Однак, у меню Edit інших GUI-модулів, використовуваних із системами нечіткого логічного виведення, додається команда Anfis..., що дозволяє відкрити ANFIS-редактор з цих модулів.

Область візуалізації містить два типи інформації: при навчанні системи – крива навчання у виді графіка залежності помилки навчання від порядкового номера ітерації; при завантаженні даних і тестуванні системи – експериментальні дані і результати моделювання.

Експериментальні дані і результати моделювання виводяться у виді множини крапок у двовимірному просторі. При цьому по вісі абсцис відкладається порядковий номер рядка даних у вибірці (навчальної, тестової або контрольної), а по осі ординат - значення вихідної змінної для даного рядка вибірки. Використовуються наступні маркери: блакитна крапка (.) – тестова вибірка; блакитна окружність (o) – навчальна вибірка; блакитний плюс (+) – контрольна вибірка; червона зірочка (*) – результати моделювання.

В *області властивостей* ANFIS (ANFIS info) виводиться інформація про кількість вхідних і вихідних змінних, про кількість функцій приналежності для кожної вхідний змінний, а також про кількість рядків у вибірках. У цій області розташовані дві кнопки Structure і Clear Plot.

Натискання кнопки Structure відчиняє нове графічне вікно, у якому система нечіткого логічного виведення представляє у виді нейро-нечіткої мережі. Натискання кнопки Clear Plot дозволяє очистити область візуалізації.

В *області завантаження даних* (Load data) розташовані: меню вибору типу даних (Type), що містить альтернативи (Traning - навчальна вибірка; Testing - тестова вибірка; Checking - контрольна вибірка; Demo - демонстраційний приклад); меню вибору джерела даних (From), що містить альтернативи (disk – диск; worksp. - робоча область MatLab); кнопка завантаження даних Load Data..., по натисканню якої з'являється діалогове вікно вибору файлу, якщо завантаження даних відбувається з диска, або вікно

введення ідентифікатора вибірки, якщо завантаження даних походить з робочої області; кнопка очищення даних Clear Data.

В області *генерування* (Generate FIS) розташовані меню вибору способу створення вихідної системи нечіткого логічного виведення. Меню містить наступні альтернативи: Load from disk – завантаження системи з диска; Load from worksp. – завантаження системи з робочої області MatLab; Grid partition - генерування системи по методу ґрат (без кластеризації); Sub. clustering – генерування системи за методом субкластеризації.

В області також розташована кнопка Generate, по натисканню якої генерується вихідна система нечіткого логічного виведення.

При виборі Load from disk з'являється стандартне діалогове вікно відкриття файлу.

При виборі Load from worksp. з'являється стандартне діалогове вікно введення ідентифікатора системи нечіткого логічного виведення.

При виборі Grid partition з'являється вікно введення параметрів методу ґрат, у якому потрібно вказати кількість термів для кожен вхідний змінної і тип функцій приналежності для вхідних і вихідної змінних.

При виборі Sub. clustering з'являється вікно введення наступних параметрів методу субкластеризації: Range of influence – рівні впливу вхідних змінних; Squash factor – коефіцієнт пригнічення; Accept ratio – коефіцієнт, що встановлює у скільки разів потенціал даної точки повинний бути вище потенціалу центра першого кластера для того, щоб центром одного з кластерів була призначена розглянута точка; Reject ratio - коефіцієнт, що встановлює у скількох разів потенціал даної точки повинний бути нижче потенціалу центра першого кластера, щоб розглянута точка була виключена з можливих центрів кластерів.

В області *навчання* (Train FIS) розташовані меню вибору методу оптимізації (Optim. method), поле завдання необхідної точності навчання (Error tolerance), поле завдання кількості ітерацій навчання (Epochs) і кнопка Train Now, натискання якого запускає режим навчання. Проміжні результати навчання виводяться в область візуалізації й у робочу область MatLab. У ANFIS-редакторі реалізовані два методи навчання: backprogra – метод зворотного поширення помилки, заснований на ідеях методу найшвидшого спуска; hybrid – гібридний метод, що поєднує метод зворотного поширення помилки з методом найменших квадратів.

В області *тестування* (Test FIS) розташовані меню вибору вибірки і кнопка Test Now, по натисканню по який відбувається тестування нечіткої системи з виведення результатів в область візуалізації.

Область виведення поточної інформації: у цій області виводиться найбільш істотна поточна інформація, наприклад, повідомлення про закінчення виконання операцій, значення помилки чи навчання тестування і т.п.

Б.7 Функції Fuzzy Logic Toolbox

Функції, що входять до модуля Fuzzy Logic Toolbox можуть бути викликані з командного рядка. Для отримання переліку функцій слід ввести команду: `help fuzzy`. Наведемо короткий огляд функцій модуля Fuzzy Logic.

Редактори з графічним інтерфейсом користувача: `anfisedit` - інструмент для навчання та тестування ANFIS; `findcluster` - інструмент для кластеризації; `fuzzy` - базовий редактор FIS; `mfedit` - редактор функцій приналежності; `ruleedit` - редактор та аналізатор правил; `ruleview` - демонстратор правил та діаграм нечіткого виведення; `surfview` - демонстратор вихідної поверхні.

Функції приналежності: `dsigmf`, `gauss2mf`, `gaussmf`, `gbellmf`, `pimf`, `psigmf`, `smf`, `sigmf`, `trapmf`, `trimf`, `zmf`.

Модуль Fuzzy Logic Toolbox пакету MATLAB включає 11 убудованих функцій приналежності, що використовують такі основні функції: кусочно-лінійну; гаусівський розподіл; сигмоїдну криву; квадратичну та кубічну криві.

Для зручності імена всіх убудованих функцій приналежності закінчуються на `mf`. Виклик функції приналежності здійснюється в такий спосіб: `namemf(x, params)`, де `namemf` – найменування функції приналежності; `x` – вектор, для координат якого необхідно розрахувати значення функції приналежності; `params` – вектор параметрів функції приналежності.

Найпростіші функції приналежності *трикутна* (`trimf`) і *трапецієподібна* (`trapmf`) формуються з використанням кусочно-лінійної апроксимації. Трапецієподібна функція приналежності є узагальненням трикутної, вона дозволяє задавати ядро нечіткої множини у виді інтервалу. У випадку трапецієподібної функції приналежності можлива наступна зручна інтерпретація: ядро нечіткої множини – оптимістична оцінка; носій нечіткої множини – песимістична оцінка.

Дві функції приналежності – *симетрична гаусівська* (`gaussmf`) і *двостороння гаусівська* (`gaussmf`) формуються з використанням гаусівського розподілу. Функція `gaussmf` дозволяє задавати асиметричні функції приналежності. Узагальнена колоколообразна функція приналежності (`gbellmf`) за своєю формою схожа на гаусівські. Ці функції приналежності часто використовуються в нечітких системах, тому що на всій області визначення вони є гладкими і приймають ненульові значення.

Функції приналежності `sigmf`, `dsigmf`, `psigmf` засновані на використанні *сигмоїдної кривої*. Ці функції дозволяють формувати функції приналежності, значення яких починаючи з деякого значення аргументу і до $+\infty$ ($-\infty$) рівні 1. Такі функції зручні для завдання лінгвістичних термів типу «високий» або «низький».

Поліноміальна апроксимація застосовується при формуванні функцій *zmf*, *pimf* і *smf*, графічні зображення яких схожі на функції *sigmf*, *dsigmf*, *psigmf*, відповідно.

У Fuzzy Logic Toolbox передбачена можливість для користувача створення власної функції приналежності. Для цього необхідно створити *m*-функцію, що містить два вхідних аргументи – вектор, для координат якого необхідно розрахувати значення функції приналежності і вектор параметрів функції приналежності. Вихідним аргументом функції повинний бути вектор ступенів приналежності.

Функції FIS: *addmf* - додає функцію приналежності до FIS; *addrule* - додає правило до FIS; *addvar* - додає змінну до FIS; *defuzz* - дефузифікує функцію приналежності; *evalfis* - здійснює обчислення нечіткого виведення; *evalmf* - обчислює функцію приналежності; *gensurf* - генерує поверхню виходу FIS; *getfis* - повертає властивості нечіткої системи; *mf2mf* - трансліує параметри між функціями приналежності; *newfis* - створює нову FIS; *parsrule* - аналізує нечіткі правила; *plotfis* - показує діаграму "вхід-вихід" для FIS; *plotmf* - показує усі функції приналежності для однієї змінної; *readfis* - завантажує FIS з диску; *rmmf* - видаляє функцію приналежності з FIS; *rmvar* - видаляє змінну з FIS; *setfis* - встановлює властивості нечіткої системи; *showfis* - показує анотовану FIS; *showrule* - відображує правила FIS; *writefis* - зберігає FIS на диску.

Функція *output=evalfis(input, fis, numofpoints)*. виконує обчислення для вибірки екземплярів за допомогою вказаної нейро-нечіткої мережі. Результатом є обчислені виходи функції, яку апроксимує нейро-нечітка мережа. Аргументи та результат функції: *input* - входи обчислюваної вибірки; *fis* - нейро-нечітка мережа; *numofpoints* - кількість точок для проведення дефазифікації (рекомендується брати значення не менше 100, зменшення цього параметру прискорює процес обчислень, але зменшує точність); *output* - обчислені виходи;

Вибірки даних та нейро-нечіткі мережі, з якими працюють описані функції повинні зберігатися в робочій області MATLAB (в оперативній пам'яті). В стовпцях матриць, що представляють собою вибірки, зберігаються значення входів (ознак) та виходів, в рядках - екземпляри вибірки.

Для завантаження вибірок з диску або запису на диск необхідно використовувати команду MATLAB: *s=load(filename)*, яка завантажує вміст файлу в змінну *s*. Функція *save(filename, s)*. зберігає змінну *s* у файл.

Для можливості використання створеної нейро-нечіткої мережі в наступних сеансах роботи або використання мережі з попередніх слід використовувати такі функції модуля Fuzzy Logic Toolbox:

`writefis(fis,filename)` - зберігає нейро-нечітку мережу `fis` в файл;
`fis=readfis(filename)` - завантажує з файлу нейро-нечітку мережу в змінну `fis`.

Передові технології: `anfis` - функція навчання для системи Сугено; функції кластер-аналізу: `fcm`, `genfis1`, `genfis2`; `subclust`.

Різні функції: `convertfis` - перетворює нечітку матрицю структури версії 1.0 у матрицю структури версії 2.0; `discfis` - дискретизує FIS; `evalmmf` - використовується для обчислення множинних функцій приналежності; `fstrvcats` - поєднує матриці різного розміру; `fuzarith` - функція нечіткої арифметики; `findrow` - шукає рядки матриці, що відповідають вхідному рядку; `genparam` - генерує початкові параметри передумов для навчання ANFIS; `probor` - імовірнісне АБО; `sugmax` - максимальний вихідний діапазон для системи Сугено.

Функція `C = fuzarith(X, A, B, OPERATOR)` реалізує базові операції нечіткої логіки та повертає нечітку множину `C` як результат застосування оператора `OPERATOR` до нечітких множин `A` та `B` з універсальної множини `X`. Змінні `A`, `B` та `X` мають бути векторами однакової розмірності. `OPERATOR` має бути одним з рядків: 'sum' - сума, 'sub' - вирахування, 'prod' - добуток, 'div' - ділення. Нечітка множина `C`, яка повертається, є вектор-стовпцем тієї ж довжини, що й `A` та `B`. Зауважимо, що ця функція використовує інтервальну арифметику та передбачає, що: `A` та `B` є опуклими нечіткими множинами; функції приналежності для `A` та `B` поза `X` є нулем.

Допоміжні функції графічного користувальницького інтерфейсу: `cmfdlg` - додає діалог вибору функцій приналежності; `cmthdlg` - додає діалог вибору методу виведення; `fisgui` - дискрипторне посилання на інтерфейсні засоби модуля Fuzzy Logic Toolbox; `gfmfdlg` - генерує FIS з використанням діалогу методу ґрат; `mfdlg` - додає діалог функції приналежності; `mfdrag` - перетягування функції приналежності за допомогою миші; `popundo` - відновлює зі стеку останні зміни (відміння останні дії); `pushundo` - передає поточні дані у стек відновлення; `savedlg` - діалог запису перед закриттям; `statmsg` - зображує повідомлення у полі статусу; `updtfis` - оновлює засоби графічного інтерфейсу Fuzzy Logic Toolbox; `wsdlg` - діалог "відкриття з" / "збереження до" робочої області.