

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ІНТЕЛЕКТУАЛЬНІ МЕТОДИ ПРОГНОЗУВАННЯ  
СОЦІАЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ**

**Методичні рекомендації  
до самостійної роботи студентів  
спеціальності 051 "Економіка"  
другого (магістерського) рівня**

**Харків  
ХНЕУ ім. С. Кузнеця  
2019**

УДК 338.27(07.034)

I-73

**Укладачі:** К. А. Стрижиченко  
С. О. Степурина

Затверджено на засіданні кафедри економічної теорії, статистики та прогнозування.

Протокол № 2 від 01.09.2018 р.

*Самостійне електронне текстове мережеве видання*

I-73 **Інтелектуальні** методи прогнозування соціально-економічних процесів [Електронний ресурс] : методичні рекомендації до самостійної роботи студентів спеціальності 051 "Економіка" другого (магістерського) рівня / уклад. К. А. Стрижиченко, С. О. Степурина. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 44 с.

Подано завдання до самостійної роботи та методичні рекомендації до їхнього виконання. Наведено індивідуальні завдання та приклади їхнього вирішення.

Рекомендовано для студентів спеціальності 051 "Економіка" другого (магістерського) рівня.

**УДК 338.27(07.034)**

© Харківський національний економічний університет імені Семена Кузнеця, 2019

# Вступ

Економічні системи, що вивчаються сучасною наукою, з великими труднощами піддаються дослідженню звичайними (вербальними) теоретичними методами. Прямий експеримент над ними неможливий. Ціна помилок і прорахунків велика, тому математичне моделювання є неминучою складовою науково-технічного прогресу.

Моделювання – основний специфічний метод наук, що застосовується для аналізу та синтезу систем управління. Це особовий пізнавальний спосіб, коли суб'єкт дослідження замість безпосереднього досліджуваного об'єкта пізнання обирає чи створює подібний до нього допоміжний об'єкт – образ чи модель, досліджує його, а отримані нові знання переносить на об'єкт – оригінал. Завдяки активній ролі суб'єкта, сам процес моделювання має творчий, активний характер.

**Метою** викладання навчальної дисципліни "Інтелектуальні методи прогнозування соціально-економічних процесів" є формування теоретичних знань та практичних навичок із питань моделювання динамічних економічних процесів. У центрі уваги перебувають методи і моделі аналізу тенденцій та причинно-наслідкових зв'язків в економіці, що є необхідною умовою аналізу й прогнозування окремих показників динаміки соціально-економічного розвитку країни.

Основні завдання вивчення дисципліни:

ознайомитися з можливостями і набути практичних навичок використання інтелектуальних методів дослідження соціально-економічних (СЕС);

ознайомитися з можливостями і набути практичних навичок використання методів нечіткої логіки;

ознайомитися з можливостями і набути практичних навичок використання нейронних мереж;

ознайомитися з можливостями і набути практичних навичок використання генетичних алгоритмів.

Згідно з вимогами освітньо-професійної програми студенти повинні:

**знати:**

основні поняття і завдання щодо використання інтелектуальних методів прогнозування СЕС;

методи нечіткої логіки;

методи нейронних мереж;

генетичні алгоритми;

**вміти:**

будувати функції приналежності;

використовувати нейронні мережі для аналізу статистичної інформації;

будувати генетичні алгоритми.

**Самостійна робота студента (СРС)** – це форма організації навчального процесу, за якої заплановані завдання студент виконує самостійно під методичним керівництвом викладача.

**Мета СРС** – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

У ході самостійної роботи студент має перетворитися на активного учасника навчального процесу, навчитися свідомо ставитися до оволодіння теоретичними і практичними знаннями, вільно орієнтуватися в інформаційному просторі, нести індивідуальну відповідальність за якість власної професійної підготовки. СРС містить: опрацювання лекційного матеріалу; опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами дисципліни; підготовку до ситуаційних занять; підготовку до виступу на семінарських заняттях; поглиблене опрацювання окремих лекційних тем або питань; пошук (підбір) та огляд літературних джерел на задану проблематику дисципліни; аналітичний розгляд наукових публікацій; контрольну перевірку студентами особистих знань за запитаннями для самодіагностики; підготовку до тестових контрольних робіт; підготовку до модульного контролю; систематизацію вивченого матеріалу з метою підготовки до семестрового екзамену.

Необхідним елементом успішного засвоєння матеріалу навчальної дисципліни є самостійна робота студентів із вітчизняною та закордонною спеціальною економічною літературою, нормативними актами з питань державного регулювання економіки, статистичними матеріалами.

Ця методична розробка вміщує завдання за такими темами: "Змінні в нечіткій логіці", "Використання алгоритмів у нечіткій логіці", "Парадигми навчання в нейронних мережах та алгоритми", "Алгоритми навчання нейронної мережі".

# **Запитання для самодіагностики за модулями та темами**

## **Змістовий модуль 1**

### **Методи нечіткої логіки в аналізі статистичної інформації**

#### **Тема 1. Загальні поняття нечіткої логіки, нечіткі множини та операції над ними**

1. Етапи еволюції нечіткої логіки.
2. Що таке функція приналежності?
3. Що таке нечітка множина?
4. У чому полягає різниця між алгебраїчними та логічними операціями над нечіткими множинами?
5. Які існують логічні операції над нечіткими множинами?
6. Які існують алгебраїчні операції над нечіткими множинами?
7. Що таке доповнення?

#### **Тема 2. Побудова функцій приналежності**

1. Які існують методи побудови функції приналежності?
2. У чому полягають особливості прямого методу?
3. Які існують переваги непрямого методу?
4. Як здійснюється експериментальний метод?
5. У чому полягає сутність методу за типовими функціями?
6. Умови використання методів побудови функції приналежності.
7. Які існують функції приналежності?
8. У чому полягають особливості побудови функції приналежності?

#### **Тема 3. Змінні в нечіткій логіці**

1. Що таке змінні в нечіткій логіці?
2. Які існують типи змінних?
3. Що таке лінгвістичні змінні?
4. Класи лінгвістичних змінних.
5. Використання змінних у нечіткій логіці.

#### **Тема 4. Використання алгоритмів у нечіткій логіці**

1. Загальне поняття алгоритмів у нечіткій логіці.
2. Алгоритми отримання нечіткого висновку.
3. Що таке дефазифікація?
4. Які існують методи дефазифікації?
5. Основні етапи алгоритмів у нечіткій логіці.
6. Фазифікація часових рядів.
7. Які існують основні фази?

### **Змістовий модуль 2**

#### **Використання нейронних мереж для дослідження соціально-економічних явищ**

##### **Тема 5. Введення в теорію нейронних мереж**

1. Що таке нейронні мережі?
2. Еволюція теорії нейронних мереж.
3. Основні поняття нейронних мереж.
4. Складові нейронних мереж.
5. Поняття персептрону.
6. Структура нейрону.
7. Вагомість персептрону.
8. Що таке аксон та синапс?

##### **Тема 6. Парадигми навчання в нейронних мережах та алгоритми**

1. Що таке парадигма навчання в нейронних мережах?
2. Які існують основні парадигми навчання?
3. Супервізорне навчання.
4. Несупервізорне навчання.
5. Посилене навчання.
6. Класифікація алгоритмів навчання.
7. Як здійснюється регулювання впливу нейрону
8. У чому сенс правила Хебба?
9. Якою є особливість корекції за похибкою?
10. У чому полягає метод конкуренції;
11. Що таке машина Больцмана?

## **Тема 7. Топології нейронної мережі**

1. Класифікація нейронних мереж.
2. Що таке топологія?
2. Прості одношарові мережі.
3. Що таке лінійний розділ?
4. Простий перцептрон.
5. Які особливості мережі Хебба?
6. Кроки алгоритму навчання перцептрону.
7. Багатошарові нейронні мережі.

## **Тема 8. Алгоритми навчання мережі**

1. Що таке алгоритм навчання системи?
2. Які існують кроки алгоритму?
3. У чому переваги простого проходження по мережі?
4. Що таке зворотний перехід по системі?
5. У чому полягає зміна важелів?
6. Які існують методи подання даних?
7. Градієнтні алгоритми навчання.

## **Тема 9. Приклади нейронних мереж**

1. Що таке подібність?
2. Що таке кодування?
3. У чому полягає сутність головних компонент?
4. Що таке кластеризація?
5. У чому полягає правило Ойя?
6. Що таке Карті Кохонена?
7. У чому переваги карт Кохонена порівняно з іншими мережами.
8. Що таке самоорганізація?

## **Тема 10. Оцінка якості нейронних мереж**

1. Перевірка якості відібраних змінних.
2. Показники оцінки якості нейронної мережі.
3. У чому полягає ефективність роботи мережі?
4. Що таке стійкість системи?
5. Що таке ефективність моделі?

## **Змістовий модуль 3**

### **Генетичні алгоритми в економіці**

#### **Тема 11. Еволюція та генетичні алгоритми**

1. Поняття еволюції.
2. Різновиди еволюції.
3. Розвиток методів еволюційних обчислень.
4. Що таке генетика?
5. Використання генетики в економіці.
6. Що таке генетичний алгоритм?
7. Основні поняття генетичних алгоритмів.

#### **Тема 12. Кодування в генетичних алгоритмах**

1. Концепції кодування в генетичних алгоритмах.
2. Види кодування.
3. Принципи кодування.
4. Фундаментна теорема ГА.
5. Що таке кодування в генетичних алгоритмах?

#### **Тема 13. Генетичні оператори**

1. Що таке генетичний оператор?
2. Сутність поняття селекції.
3. Схрещування.
4. Поняття мутації.
5. У чому полягають принципи мутації?
6. Поєднання генетичних операторів.
7. Використання генетичних операторів на фінансовому ринку.

### **Завдання для самостійної роботи за темою 3**

### **"Проектування систем нечіткого висновку на основі алгоритму Мамдані"**

**Мета роботи:** освоїти методику проектування системи нечіткого висновку на основі розробки та використання баз знань продукційних правил з використанням алгоритму Мамдані.



### Завдання для самостійного виконання за темою 3

1. Створити систему нечіткого висновку, яка моделює залежність  $y = x_1^2 \sin(x_2 - 1)$  при  $x_1 \in [-7,3]$ ;  $x_2 \in [-4.5,2]$ .

2. Створити систему нечіткого висновку, яка відтворює поверхню  $y = \ln(x_1 + 1)\cos(x_2)$  при  $x_1 \in [1,5]$ ;  $x_2 \in [0,2]$ .

3. Створити систему нечіткого висновку, яка відтворює поверхню  $y = \ln(x_1 + 1)\frac{1}{1+x_2}$  при  $x_1 \in [1,5]$ ;  $x_2 \in [0,2]$ .

#### Методичні рекомендації до виконання завдання

**Основні поняття.** Знання можна формалізувати у вигляді системи нечітких логічних висловлювань. Кожне висловлювання можна оцінити нечітким ступенем істинності. Наприклад, висловлювання "швидкість машини висока" може бути істинне на 80 %, а висловлювання "завтра буде морозна погода" – на 100 %. Кожне таке висловлювання можна описати за допомогою відношень множин лінгвістичних нечітких змінних.

**Лінгвістична змінна** – це кортеж наступних значень  $\langle \beta, T, X, G, M \rangle$ , де  $\beta$  – ім'я змінної (наприклад, "швидкість автомобіля");

$T$  – базова множина значень її термів – значень, кожне з яких надається за допомогою нечіткої множини (наприклад, "мала", "середня", "висока", "дуже висока");

$X$  – множина – носій можливих конкретних значень змінної для всіх термів (наприклад,  $X = [0,200]$  км/год.);

$G$  – деяка синтетична процедура генерації нових термів із множини  $T$  (наприклад, "дуже мала");

$M$  – семантична процедура надання терму певної нечіткої змінної вигляду  $\langle X, \mu_i(X) \rangle$ ,  $\mu_i(X)$  – функція приналежності  $i$ -того терму з множини  $T$ .

У системі **MATLAB** існує середовище для формування систем знань нечіткого висновку. Для входу в це середовище слід ввести в командному рядку слово **fuzzy** і натиснути клавішу **<Enter>**. Побудова системи нечіткого висновку (СНВ), яка основана на використанні алгоритму Мамдані, має наступні етапи:

1. Проектування бази правил СНВ. Кожне правило представляється у вигляді:

**Якщо** <умова> **тоді** <заключення> [міра правильності правила]

Для алгоритму Мамдані <умова> і <заключення> виглядають як логічні зв'язки наступних записів: **<нечітка змінна> = < значення >**.

2. Введення цих правил в СНВ.

3. Використання СНВ для обробки вхідної інформації у вигляді конкретних значень вхідних (нечітких) змінних. Цей етап, своєю чергою, розкладається на наступні складові:

3.1. Введення значень вхідних змінних. Тобто деяких фактів, які вважаються істинними на 100 %.

3.2. Фазифікація вхідних змінних – встановлення відповідності між конкретним значенням вхідних змінних і значенням її терму, разом з функцією приналежності.

3.3. Агрегування складних умов, які стоять в правилах після ключового слова ЯКЩО, тобто визначення степені істинності всіх умов в усіх правилах, якщо умови надаються за допомогою складних логічних виразів. Правило активується, якщо істинність його умови більша за нуль. У базах знань процедура агрегування умов в правилах виконується за допомогою нечітких логічних операцій – нечіткої кон'юнкції, нечіткої диз'юнкції, нечіткої відмови, та ін.

3.4. Активація підзаключень – процес визначення ступеня істинності (приналежності до відповідних термів) змінних, які посідають у формуваннях активних правил, за формулою:  $c_k = b_k F_k$ , де  $c_k$  – ступінь істинності формування правила  $k$ ,  $b_k$  – ступінь істинності його умови,  $F_k$  – ступінь істинності самого правила (ваговий коефіцієнт  $k$  – правила). Після визначення вектора  $C = (c_1, \dots, c_q)$  визначаються функції приналежності для кожного із підзаключень для кожної вихідної лінгвістичної змінної. Припустимо, що відповідний терм вихідної лінгвістичної змінної визначається функцією приналежності  $\mu(y)$ . Тоді після процедури активації отримуємо поновлену функцію приналежності відповідного терма (підзаключення)  $\mu'(y)$  за одним із методів нечіткої композиції:

min – активізація:  $\mu'(y) = \min\{c_i, \mu(y)\}$ ;

prod – активізація:  $\mu'(y) = c_i \mu(y)$ ;

average – активізація:  $\mu'(y) = 0.5(c_i + \mu(y))$ .

Відзначимо, що різні правила підзаключень можуть містити однакові терми лінгвістичних змінних. У цьому випадку для кожного терма ми визначаємо множину різних функцій приналежності, які обчислюються за одним із правил нечіткої композиції за кожним правилом продукцій. Остаточна функція приналежності для цього терма визначається в наступному пункті.

3.5. Акумуляція заключень, тобто, визначення значення функцій приналежності для термів всіх вихідних змінних. Якщо для одного терму визначена множина функцій приналежності  $\mu'_1(y), \dots, \mu'_p$ , то акумуляція виконується за одним із правил об'єднання нечітких множин:

об'єднання:  $\mu'(y) = \max\{\mu'_1(y), \mu'_2\}$ ;

алгебраїчне об'єднання:  $\mu'(y) = \mu'_1(y) + \mu'_2(y) - \mu'_1(y)\mu'_2(y)$ ;

граничне об'єднання:  $\mu'(y) = \max\{\mu'_1(y) + \mu'_2(y) - 1, 0\}$ ;

операція  $\lambda$  – суми:  $\mu'(y) = \lambda\mu'_1(y) + (1 - \lambda)\mu'_2(y)$ ,  $\lambda \in [0,1]$ .

драстичне об'єднання:  $\mu'(y) = \begin{cases} \mu'_1(y), & \text{if } \mu'_2(y) = 0, \\ \mu'_2(y), & \text{if } \mu'_1(y) = 0, \\ 1, & \text{else.} \end{cases}$

3.6. Дефазифікація вихідних змінних (визначення конкретних значень за функціями приналежності термів) розглядається методом центру ваги для неперервних та дискретних нечітких множин за формулами:

$$z = \frac{\int_{y_{\min}}^{y_{\max}} y\mu'(y)dy}{\int_{y_{\min}}^{y_{\max}} \mu'(y)dy}, \quad z = \frac{\sum_{i=1}^n y_i\mu'(y_i)}{\sum_{i=1}^n \mu'(y_i)}.$$

Розглянемо принципи побудови та роботи системи нечіткого висновку на прикладі задачі візуалізації поверхні, яка задана функцією, тобто проектування та використання системи нечіткого висновку.

Наприклад, за допомогою СНВ зобразити поверхню функції  $y = (x_1^2 - 8)\cos(x_2)$  на множині  $x_1 \in [0,4]; x_2 \in [0,4]$ .

Проектування системи нечіткого висновку слід проводити на основі графічного зображення вказаної залежності. Для цього в *M*-файлі складемо наступну програму:

Побудова графіка функції  $y=(x_1^2-8)*\cos(x_2)$   
в області  $x_1 \in [0,4]$  и  $x_2 \in [0,4]$ .

```
n=15;  
x1=0:4/(n - 1):4;  
x2=0:4/(n - 1):4;  
y=zeros(n,n);  
for j=1:n  
y(j,:)=(x1.^2 - 8)*cos(x2(j));  
end  
surf(x1,x2,y)  
xlabel('x1')  
ylabel('x2')  
zlabel('y')  
title('Target');
```

У результаті виконання цієї програми отримуємо графічне зображення, яке наведено на рис. 1.

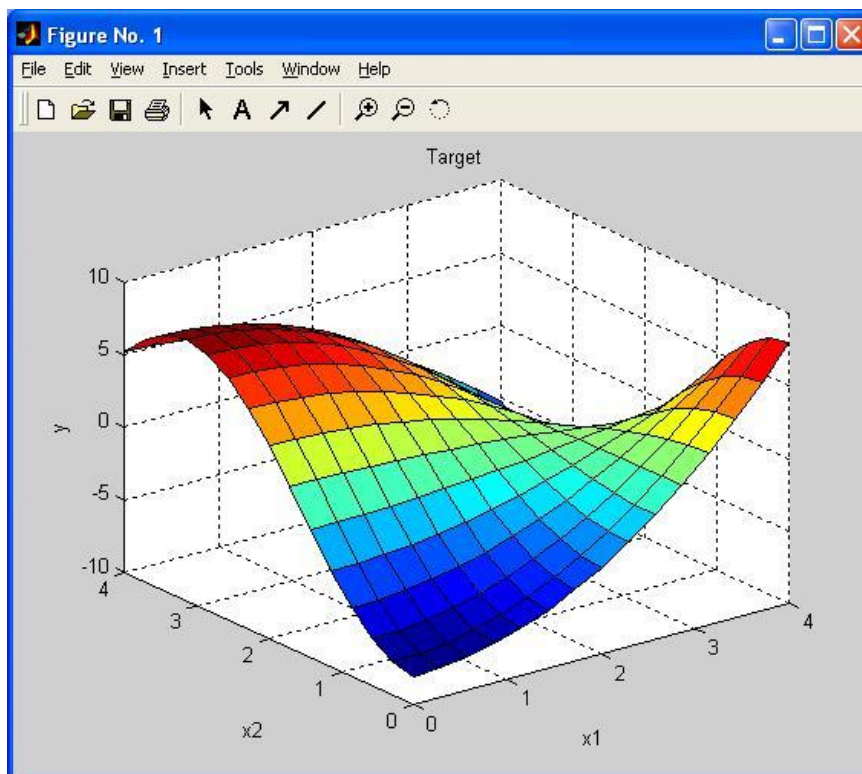


Рис. 1. Зображення поверхні залежності

Проектування СНВ складається з наступних кроків.

*Крок 1.* Завантажити основний fis-редактор в редактор нечіткого висновку введенням в командному рядку слова fuzzy. Після чого з'явиться вікно редактора нечіткого висновку.

*Крок 2.* Ввести нову вхідну змінну. Для цього вибрати пункт Add Input в меню Edit.

*Крок 3.* Перейменувати першу вхідну змінну. Для цього слід зробити одне натиснення лівої кнопки миші на блоці Input1, ввести нове позначення x1 в поле редагування імені поточної змінної і натиснути <Enter>.

*Крок 4.* Перейменувати другу вхідну змінну. Для цього зробити одне натиснення лівої кнопки миші на блоці input2, ввести нове позначення x2 в поле редагування імені поточної змінної і натиснути <Enter>.

*Крок 5.* Перейменувати вихідну змінну. Для цього зробити одне натиснення лівою кнопкою миші на блоці output1, ввести нове позначення у в поле редагування імені поточної змінної і натиснути <Enter>.

*Крок 6.* Задати ім'я системі. Для цього в меню File вибрати в підменю Export to disk і ввести ім'я файлу, наприклад, first.

*Крок 7.* Перейти в редактор функцій приналежності. Для цього зробити швидко подвійне натиснення лівої кнопки миші на блоці x1.

*Крок 8.* Задати діапазон зміни змінної x1. Для цього надрукувати 0 4 в поле Range і натиснути <Enter>.

*Крок 9.* Задати функції приналежності змінної x1. Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями приналежності. Якщо у вікні немає ще функцій приналежності, тоді в меню Edit слід вибрати команду Add MFs... У результаті з'явиться діалогове вікно вибору типу і кількості функцій приналежності. За замовченням ці 3 терми мають трикутну функцію приналежності. Тому просто потрібно натиснути <Enter>.

*Крок 10.* Задати найменування термів змінної x1. Для цього робимо одне натиснення лівою кнопкою миші на графіку першої функції приналежності. (рис. 2). Потім вводимо найменування терму, наприклад, L (Низький), в полі Name і натискаємо <Enter>. Потім робимо одне натиснення лівою кнопкою миші на графіку другої функції приналежності і вводимо найменування терму, наприклад, A (Середній), в полі Name і натискаємо <Enter>. Ще раз робимо одне натиснення лівою кнопкою миші по графіку третьої функції приналежності і введемо найменування терму, наприклад, H (Високий), в полі Name і натискаємо <Enter>. У результаті отримуємо графічне вікно, яке зображено на рис. 2.

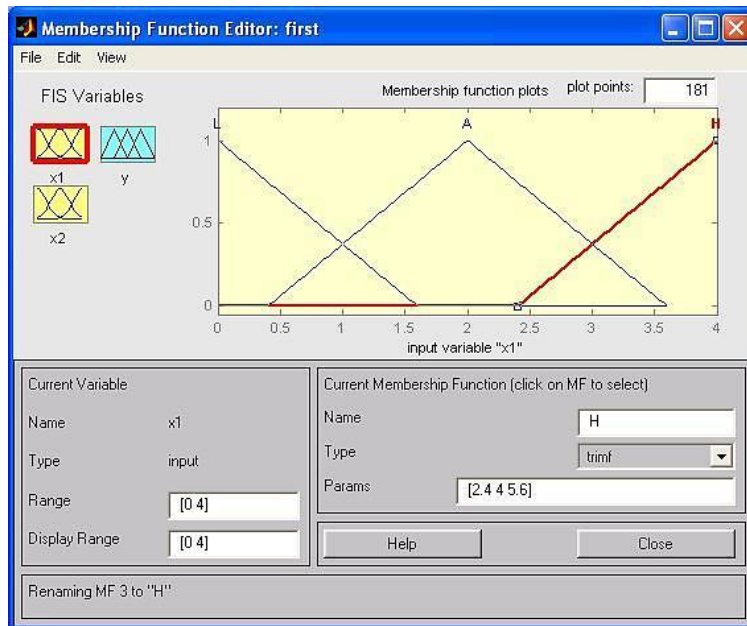


Рис. 2. Функція приналежності змінної  $x_1$

*Крок 11.* Задамо функції приналежності змінної  $x_2$ . Для лінгвістичної оцінки цієї змінної будемо використовувати 5 термів з гаусовськими функціями приналежності. Для цього активізуємо змінну  $x_2$  за допомогою натиснення лівої кнопки миші на блоці  $x_2$ . Задамо діапазон змін  $x_2$ . Для цього надрукуємо 0 4 в полі Range (рис. 3) і натиснемо <Enter>. Потім у меню Edit виберемо команду Add MFs.... У діалоговому вікні, що з'явиться, оберемо тип функції приналежності `gaussmf` в полі MF type і 5 термів у полі Number of MFs. Після чого натискаємо <Enter>.

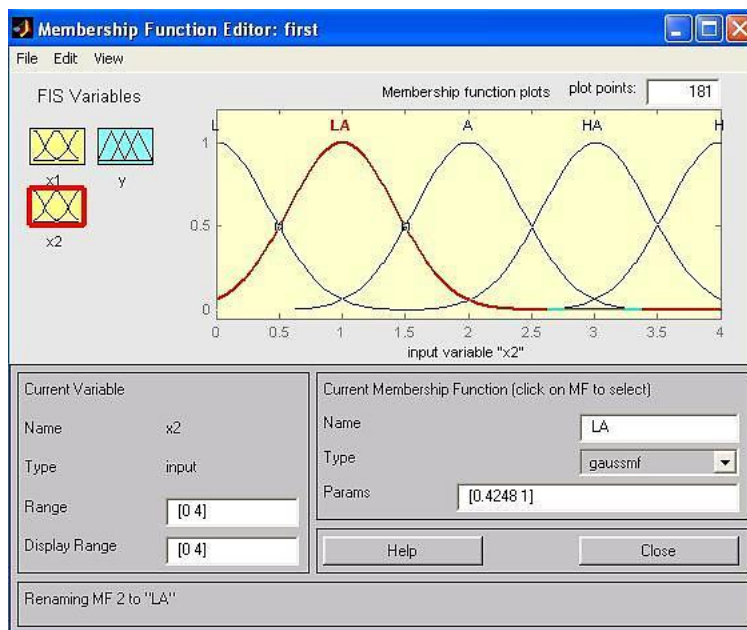
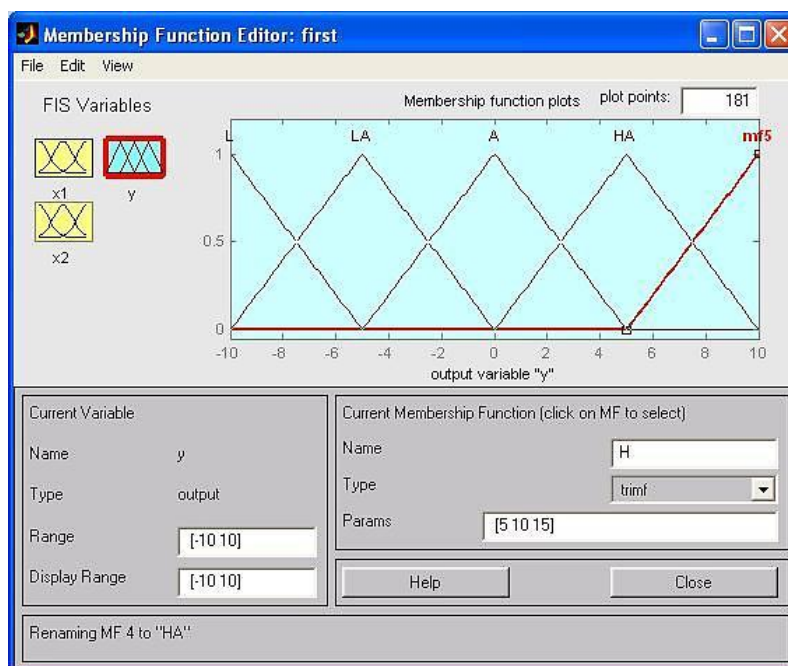


Рис. 3. Функція приналежності змінної  $x_2$

*Крок 12.* За аналогією з кроком 10 задамо наступні найменування термів змінної  $x_2$ : L (Низький), LA (Нижче середнього), A (Середній), HA (Вище середнього), H (Високий). У результаті отримуємо графічне вікно, яке зображене на рис. 3.

*Крок 13.* Задамо функції приналежності змінної  $y$ . Для лінгвістичної оцінки цієї змінної будемо використовувати 5 термів із трикутними функціями приналежності. Для цього активуємо змінну  $y$  за допомогою натиснення лівої кнопки миші на блоці  $y$ . Задамо діапазон змін змінної  $y$ . Для цього надрукуємо -10 10 в полі Range (рис. 4) і натиснемо <Enter>. Потім в меню Edit оберемо команду Add MFs.... У діалоговому вікні, що з'явиться, виберемо 5 термів в полі Number of MFs. Після чого натискаємо <Enter>.

*Крок 14.* За аналогією з кроком 10 задамо наступні найменування термів змінної  $y$ : L (Низький), LA (Нижче середнього) A (середній), HA (Вище середнього), H (Високий). В результаті отримуємо графічне вікно, яке подано на рис. 4.



**Рис. 4. Функції приналежності змінної  $y$**

*Крок 15.* Перейдемо в редактор бази знань RuleEditor. Для цього оберемо в меню Edit команду Rules або в меню View команду Edit rules....

*Крок 16.* На основі візуального спостереження за графіком, який зображений на рис. 1, сформуємо наступні десять правил (рис. 5):

1. Якщо  $x_1$ =Низький і  $x_2$ =Низький, тоді  $y$ =Низький;
2. Якщо  $x_1$ =Низький і  $x_2$ =Високий, тоді  $y$ =Середній;

3. Якщо  $x_1$ =Низький і  $x_2$ =Вище середнього, тоді  $y$ =Високий;
4. Якщо  $x_1$ =Високий і  $x_2$ =Низький, тоді  $y$ =Вище середнього;
5. Якщо  $x_1$ =Високий і  $x_2$ =Високий, тоді  $y$ =Низький;
6. Якщо  $x_1$ =Середній і  $x_2$ =Середній, тоді  $y$ =Середній;
7. Якщо  $x_1$ =Середній і  $x_2$ =Вище середнього, тоді  $y$ =Вище середнього;
8. Якщо  $x_1$ =Низький і  $x_2$ =Нижче середнього, тоді  $y$ =Нижче середнього;
9. Якщо  $x_1$ =Середній і  $x_2$ =Вище середнього, тоді  $y$ =Середній.

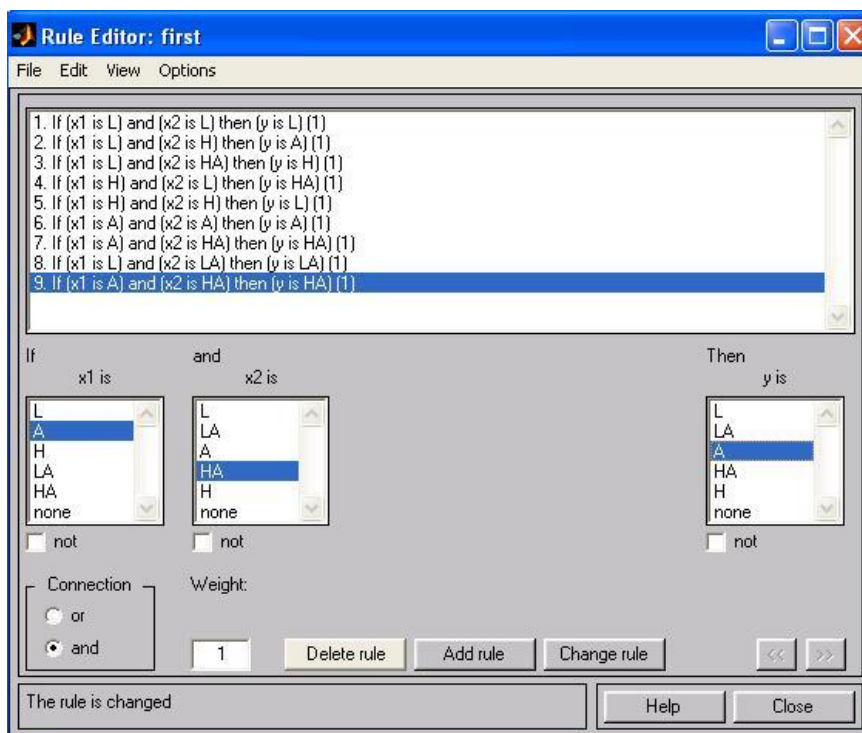


Рис. 5. Вікно правил

Для введення правила необхідно обрати в меню відповідну комбінацію термів і натиснути кнопку Add rule. На рис. 5 зображено вікно редактора бази знань після введення усіх 9 правил. Число в дужках в кінці кожного правила являє собою вагові коефіцієнти відповідного правила. Потрібно звернути увагу на параметр Weight, який вказує вагу нечіткої впевненості в правилі. Його можна задавати в діапазоні  $[0, 1]$ .

*Крок 17.* Збережемо побудовану систему. Для цього в меню File в підменю Export оберемо команду To disk.

На рис. 6 наведено вікно візуалізації нечіткого логічного висновку. Це вікно активується командою View rules... меню View. У полі Input вказуються значення вхідних змінних, для яких виконується логічний вивід. Тобто обраховується за алгоритмом Мамдані значення вихідної змінної.



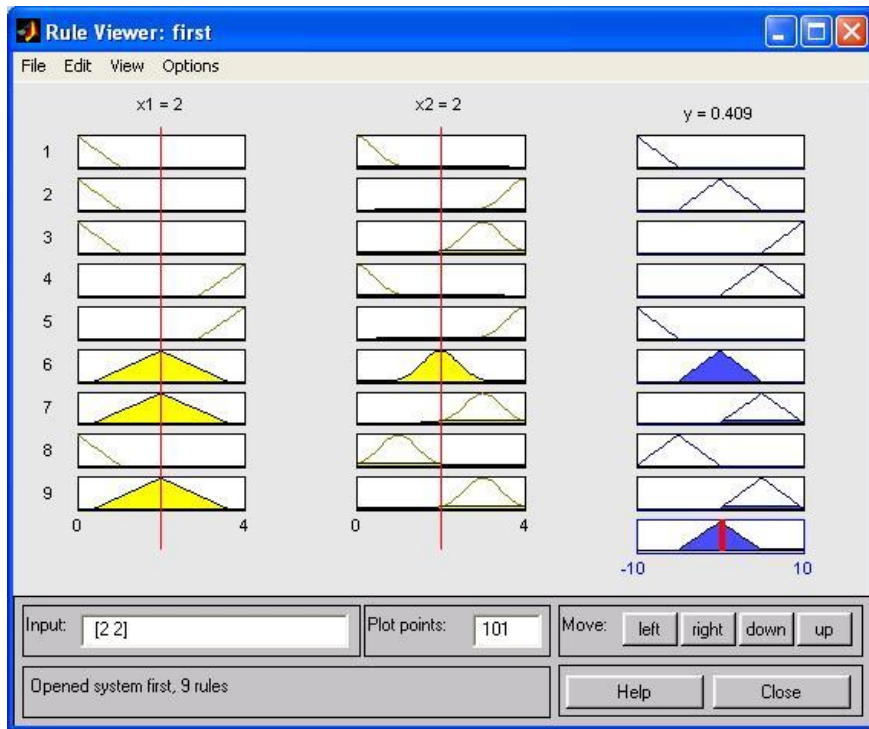


Рис 6. Візуалізація нечіткого висновку

На рис. 7 наведено поверхню "входи – вихід", яка відповідає синтезованій системі логічного висновку.

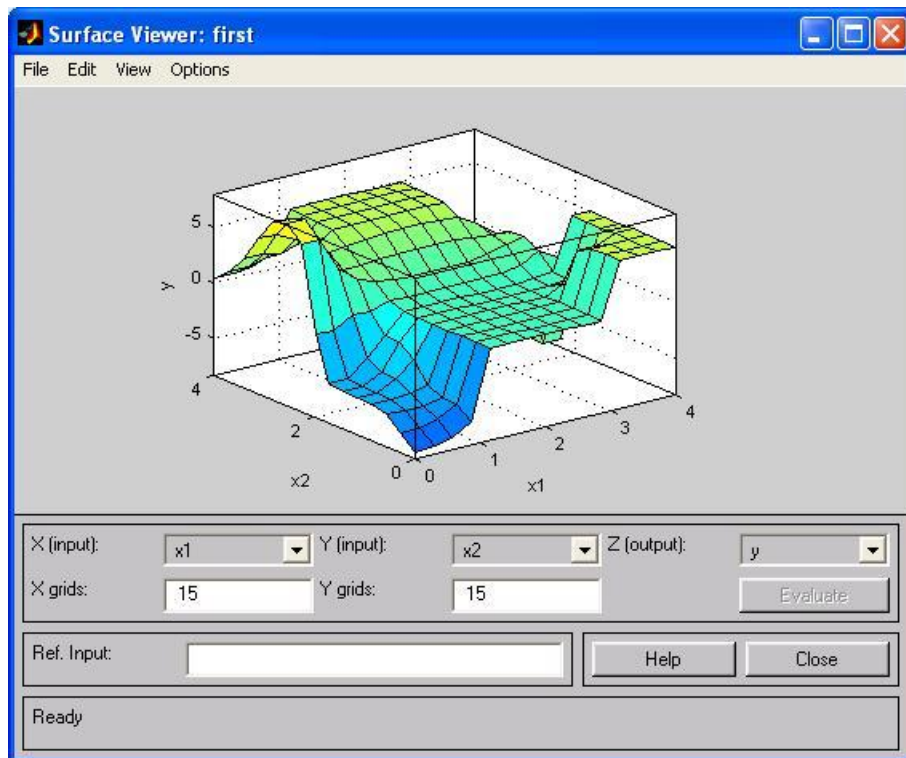


Рис. 7. Поверхня системи "вхід – вихід"

Для виведення цього вікна необхідно використати команду **View surface...** меню **View**. Порівнюючи поверхні на рис 1 та рис. 7, можна зробити висновок, що нечіткі правила досить добре описують складну нелінійну залежність.

## **Завдання для самостійної роботи за темою 6. "Розроблення інтелектуальних систем на основі моделей нейронних мереж"**

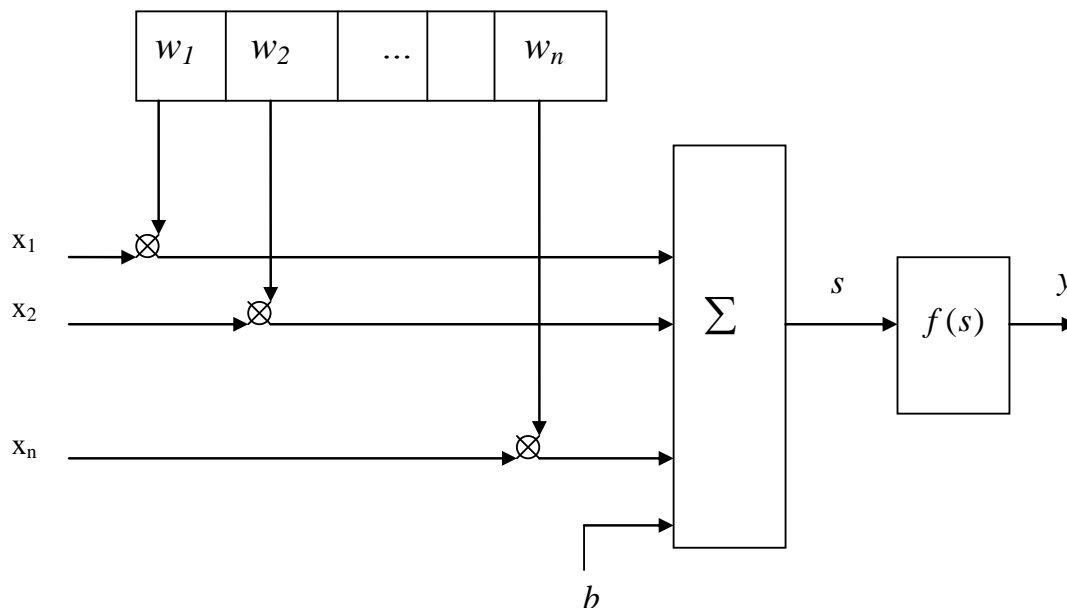
**Мета роботи:** засвоїти методику проектування і побудови інтелектуальних систем на основі моделей нечітких нейронних мереж.

### *Завдання для самостійного виконання за темою 6*

Побудувати адаптивну систему нейро-нечіткого виводу для апроксимації деякої виробничої функції типу Кобба – Дугласа.

### *Методичні рекомендації до виконання завдання*

**Теоретичні відомості.** Концептуальною основою і складовою частиною штучних нейронних мереж є так званий *штучний нейрон*, що має визначену внутрішню структуру (рис. 8) і правила перетворення сигналів.



**Рис. 8. Структура штучного нейрона**

**Штучний нейрон** (далі – нейрон) складається з помножувачів (синапсів), суматора і нелінійного перетворювача. *Синапси*, зображені перекресленим кружком, призначені для зв'язку нейронів між собою і множать вхідний сигнал  $x_i$  на деяке постійне число. Це число  $w_i$ , назване *вагою синапса*, характеризує силу цього зв'язку. Суматор виконує додавання всіх сигналів, що надходять на вхід нейрона від інших нейронів, і зовнішніх вхідних сигналів. *Нелінійний перетворювач* призначений для нелінійної зміни вихідного значення суматора відповідно до деякої функції від одного аргументу. Ця функція називається *функцією активації* чи *передатною функцією* нейрона.

Правила перетворення сигналів визначаються математичною моделлю нейрона, яка може бути записана у формі наступних аналітичних виразів:

$$\begin{aligned} s &= \sum_{i=1}^n w_i \cdot x_i + b; \\ y &= f(s), \end{aligned} \tag{1}$$

де  $w_i$  – вага синапса ( $i \in \{1, 2, \dots, n\}$ );

$b$  – значення зсуву;

$s$  – результат підсумовування;

$x_i$  – компонент вектора чи входу вхідного сигналу ( $i \in \{1, 2, \dots, n\}$ );

$y$  – вихідний сигнал нейрона;

$n$  – число входів нейрона;

$f$  – функція активації (передатна функція) нейрона, що представляє собою деяке нелінійне перетворення.

У загальному випадку:  $w_i, x_i, b \in \mathbb{R} (i \in \{1, 2, \dots, n\})$ .

Синаптичні зв'язки з додатними вагами:  $w_i \in \mathbb{R}_+ (i \in \{1, 2, \dots, n\})$  називаються збуджуючими, а з від'ємними вагами  $w_i \in \mathbb{R}_- (i \in \{1, 2, \dots, n\})$  гальмуючими.

Таким чином, окремо узятий штучний нейрон цілком описується своєю структурою (див. рис. 8) і математичною моделлю (1). Одержавши вектор вхідного сигналу  $x_i$ , нейрон видає деяке число  $y$  на своєму виході.

Як функція активації нейрона можуть бути використані різні нелінійні перетворення (табл. 1).

### Основні види функцій активації нейронів

| Назва                                | Область значень | Формула   |
|--------------------------------------|-----------------|---|
| Лінійна                              | $\mathbb{R}$    | $f(s) = k \cdot s$  |
| Напівлінійна                         | $\mathbb{R}_+$  | $f(s) = \begin{cases} 0, & \text{якщо } s \leq 0 \\ k \cdot s, & \text{якщо } s > 0 \end{cases}$                              |
| Гранична                             | $\{0, 1\}$      | $f(s) = \begin{cases} 0, & \text{якщо } s < T \\ 1, & \text{якщо } s \geq T \end{cases}$                                      |
| Модульна                             | $\mathbb{R}_+$  | $f(s) =  s $  |
| Знакова (сигнатурна)                 | $\{-1, 1\}$     | $f(s) = \begin{cases} -1, & \text{якщо } s \leq T \\ 1, & \text{якщо } s > T \end{cases}$                                     |
| Квадратична                          | $\mathbb{R}_+$  | $f(s) = s^2$  |
| Експонентна                          | $\mathbb{R}_+$  | $f(s) = e^{-as}$  |
| Синусоїдальна                        | $[-1, 1]$       | $f(s) = \sin(s)$  |
| Логістична (сигмоїдальна)            | $(0, 1)$        | $f(s) = \frac{1}{1 + e^{-as}}$  |
| Раціональна (сигмоїдальна)           | $(-1, 1)$       | $f(s) = \frac{s}{a +  s }$  |
| Гіперболічний тангенс (сигмоїдальна) | $(-1, 1)$       | $f(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$  |
| Лінійна з насиченням (крокова)       | $[-1, 1]$       | $f(s) = \begin{cases} -1, & \text{якщо } s \leq -1 \\ s, & \text{якщо } -1 < s < 1 \\ 1, & \text{якщо } s \geq 1 \end{cases}$ |
| Напівлінійна з насиченням            | $[0, 1)$        | $f(s) = \begin{cases} 0, & \text{якщо } s \leq 0 \\ s, & \text{якщо } 0 < s < 1 \\ 1, & \text{якщо } s \geq 1 \end{cases}$    |
| Трикутна                             | $[0, 1]$        | $f(s) = \begin{cases} 1 -  s , & \text{якщо }  s  \leq 1 \\ 0, & \text{якщо }  s  > 1 \end{cases}$                            |
| Радіальна базисна (Гаусова)          | $(0, 1]$        | $f(s) = e^{-s^2}$   |

Нейронна мережа являє собою сукупність окремих нейронів, взаємозв'язаних між собою деяким фіксованим чином. До того ж взаємозв'язок нейронів визначається чи задається структурою (топологією) нейронної

мережі. З точки зору топології нейронні мережі можуть бути повнозв'язними, багатошаровими і слабозв'язними. У загальному випадку *структура* багатошарової чи багаторівневої нейронної мережі може бути зображена в такий спосіб (рис. 9).

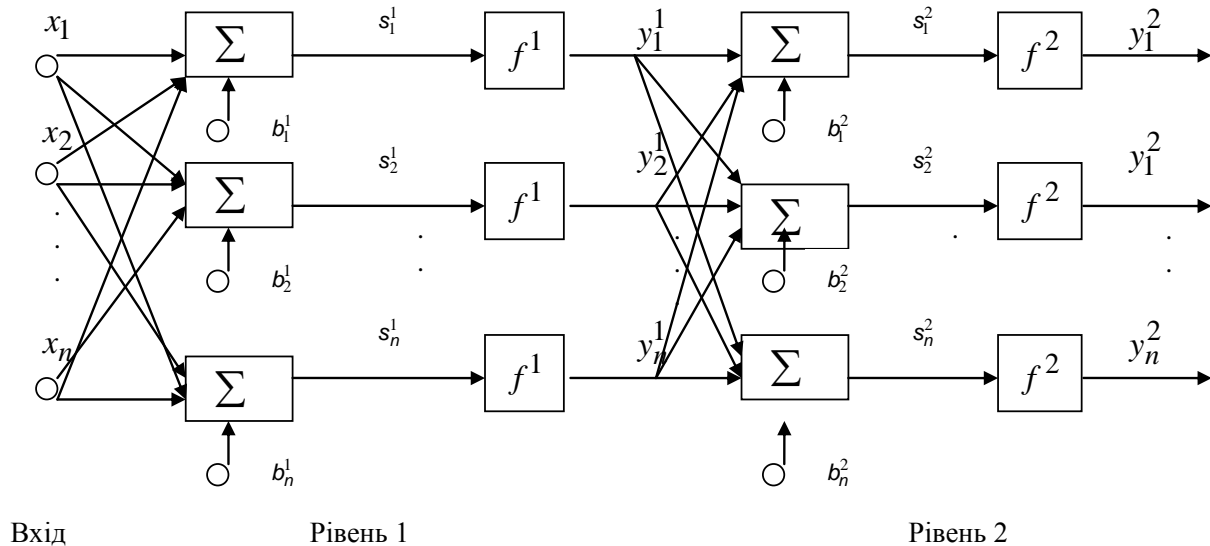


Рис. 9. Структура трирівневої нейронної мережі

Кожний із рівнів нейронної мережі називається її *шаром*. Водночас шар вхідного рівня називається *вхідним шаром*, шар рівня 1 і 2 – *схованими шарами*, а шар рівня 3 – *вихідним шаром*.

Зі свого боку багатошарові нейронні мережі можуть бути наступних типів.

Монотонні – кожен шар (крім вихідного) додатково розбивається на два блоки: збуджуючий і гальмуючий. Аналогічно розбиваються і зв'язки між блоками: на збуджуючі і гальмуючі. Водночас як функції активації можуть бути використані тільки монотонні функції (див. табл. 1).

Нейронні мережі зі зворотними зв'язками – інформація з наступних шарів може передаватися на нейрони попередніх шарів.

Нейронні мережі без зворотних зв'язків – інформація з наступних шарів не може передаватися на нейрони попередніх шарів.

Класичним варіантом багатошарової нейронної мережі є повнозв'язна мережа прямого поширення (див. рис. 9).

Процес побудови і використання нейро-мережових моделей складається з наступних етапів:

1. Вибір типу і структури нейронної мережі для розв'язку поставленої проблеми (синтез структури нейронної мережі).

2. Навчання нейронної мережі (визначення чисельних значень ваг кожного з нейронів) на основі наявної інформації про розв'язок даної задачі експертом або даних про розв'язок задачі в минулому.

3. Перевірка нейронної мережі на основі використання деякого контрольного приклада (необов'язковий етап).

4. Використання навченої нейронної мережі для розв'язку поставленої проблеми.

На даний час запропоновані різні схеми класифікації нейронних мереж і відповідні алгоритми їх навчання. Одним із найпоширеніших алгоритмів навчання є так званий алгоритм *зворотного поширення помилки* (back propagation). Цей алгоритм являє собою ітеративний градієнтний алгоритм мінімізації середньоквадратичного відхилення значень виходу від бажаних значень (мінімізації помилки) у багатошарових нейронних мережах.

Вибір виду і структури нейронної мережі визначається специфікою розв'язуваної задачі. До того ж для розв'язку окремих типів практичних задач розроблені оптимальні конфігурації нейронних мереж, що найбільше адекватно відбивають особливості відповідної проблемної області. Подальшим розвитком нейронних мереж є так звані гібридні мережі, що реалізовані в пакеті Fuzzy Logic Toolbox системи MATLAB.

*Гібридна мережа* являє собою багатошарову нейронну мережу спеціальної структури без зворотних зв'язків, у якій використовуються звичайні (не нечіткі) сигнали, ваги і функції активації, а виконання операції підсумовування засновано на використанні фіксованої Т-норми, Т-конорми чи деякої іншої неперервної операції. Водночас значення входів, виходів і ваг гібридної нейронної мережі являють собою дійсні числа з відрізка [0, 1].

Основна ідея, покладена в основу моделі гібридних мереж, полягає в тому, щоб використовувати існуючу вибірку даних для визначення параметрів функцій приналежності, що найкраще відповідають деякій системі нечіткого виводу. Водночас для знаходження параметрів функцій приналежності використовуються відомі процедури навчання нейронних мереж.

У пакеті Fuzzy Logic Toolbox системи MATLAB гібридні мережі реалізовані у формі так званої адаптивної системи нейро-нечіткого виведення ANFIS. З одного боку, гібридна мережа ANFIS являє собою нейронну мережу з єдиним виходом і декількома входами, що являють собою нечіткі лінгвістичні змінні. Водночас терми вхідних лінгвістичних змінних описуються стандартними для системи MATLAB функціями приналежності, а терми вихідної змінної представляються лінійною чи постійною функцією приналежності.

З іншого – гібридна мережа ANFIS являє собою систему нечіткого виведення FIS типу Сугено нульового чи першого порядку, у якій кожне з правил нечітких продукцій має постійну вагу, рівну 1. У системі MATLAB користувач має можливість редагувати та налаштовувати гібридні мережі ANFIS аналогічно системам нечіткого виведення, використовуючи всі розглянуті раніше засоби пакета Fuzzy Logic Toolbox.

У пакеті Fuzzy Logic Toolbox системи MATLAB гібридні мережі реалізовані у формі адаптивних систем нейро-нечіткого виведення ANFIS. Водночас розроблення і дослідження гібридних мереж виявляється можливим у разі:

інтерактивного режиму за допомогою спеціального графічного редактора адаптивних мереж, що одержав назву редактора ANFIS;

режиму командного рядка за допомогою введення імен відповідних функцій із необхідними аргументами безпосередньо у вікно команд системи MATLAB. Для роботи в режимі командного рядка призначені спеціальні функції.

Редактор ANFIS дозволяє створювати чи завантажувати конкретну модель адаптивної системи нейро-нечіткого виведення, виконувати її навчання, візуалізувати її структуру, змінювати і налагоджувати її параметри, а також використовувати налагоджену мережу для одержання результатів нечіткого виведення.

Графічний інтерфейс редактора ANFIS викликається функцією **anfisedit** з командного рядка середовища Matlab.

Для ілюстрації процесу розробки гібридної мережі в системі MATLAB розглянемо задачу побудови адаптивної системи нейро-нечіткого виведення для апроксимації деякої виробничої функції типу Кобба – Дугласа:

$$q = F(x_1, x_2) = 2\sqrt{x_1} 4\sqrt{x_2},$$

яка описує залежність між витратами двох виробничих факторів (наприклад, грошей і праці) у кількості  $x_1$  і  $x_2$  та кількістю виготовленої продукції  $q$  у наперед визначених одиницях.

Загальна послідовність процесу розробки моделі гібридної мережі може бути представлена в наступному вигляді.

*Крок 1.* Для початку за допомогою редактора відладчика m-файлів або будь-якого текстового редактору підготуємо навчальні дані, що містять

6-ти рядків з трьох значень – два вхідних і одне вихідне значення, наприклад, наступного виду:

|   |   |      |
|---|---|------|
| 1 | 2 | 2.37 |
| 4 | 1 | 4.00 |
| 1 | 4 | 2.82 |
| 5 | 5 | 6.68 |
| 2 | 4 | 4.00 |
| 4 | 3 | 5.26 |
| 3 | 1 | 3.46 |
| 4 | 4 | 5.65 |
| 4 | 2 | 4.80 |

Числа в рядку розділяються клавішею "Space", а перехід на наступний рядок відбувається натисненням клавіші "Enter". Слід зберегти цей файл у папці "Work" каталогу Matlab, наприклад, з назвою train.txt.

Взагалі вхідні дані представляють собою звичайну числову матрицю розмірності  $m \times (n+1)$ , у якій кількість рядків  $m$  відповідає обсягу вибірки, перші  $n$  стовпців – значенням вхідних змінних моделі, а останній стовпчик – значенню вихідної змінної. Відповідно до правил системи MATLAB окремі значення матриці відокремлюються пробілами, а кожен рядок матриці завершується символом "*перевід каретки*" (клавіша <Enter>).

Хоча за кількістю рядків матриці вхідних даних не існує формальних рекомендацій, прийнято вважати, що якість навчання гібридної мережі, а, отже, і точність одержуваних результатів пропорційно залежить від обсягу навчальної вибірки. Що стосується кількості стовпців матриці вхідних даних, то слід зазначити можливі проблеми з працездатністю системи MATLAB, якщо кількість вхідних змінних перевищує 5 – 6.

Початкові дані, що завантажуються, можуть бути одного з наступних типів:

навчальні дані (Training) – обов'язкові дані, що використовуються для побудови гібридної мережі;

тестові дані (Testing) – необов'язкові дані, що використовуються для тестування побудованої гібридної мережі з метою перевірки якості функціонування побудованої гібридної мережі;

перевірочні дані (Checking) – необов'язкові дані, що використовуються для перевірки побудованої гібридної мережі з метою з'ясування факту перенавчання мережі;

демонстраційні дані (Demo) – дозволяють завантажити один із демонстраційних прикладів гібридної мережі.



Тому слід створити ще і файл тестових даних test.txt. За аналогією зі створенням файлу train.txt, введемо в нього наступні дані:

|   |   |      |
|---|---|------|
| 2 | 1 | 2.82 |
| 3 | 2 | 4.12 |
| 1 | 3 | 2.63 |
| 2 | 3 | 3.72 |
| 3 | 3 | 4.56 |
| 4 | 2 | 4.75 |
| 3 | 4 | 4.90 |

Крок 2. Головне меню редактора ANFIS достатньо просте і призначене для роботи з попередньо створеною системою нечіткого виведення. Основну частину графічного інтерфейсу займає вікно візуалізації даних, що розташоване нижче головного меню (рис. 10).

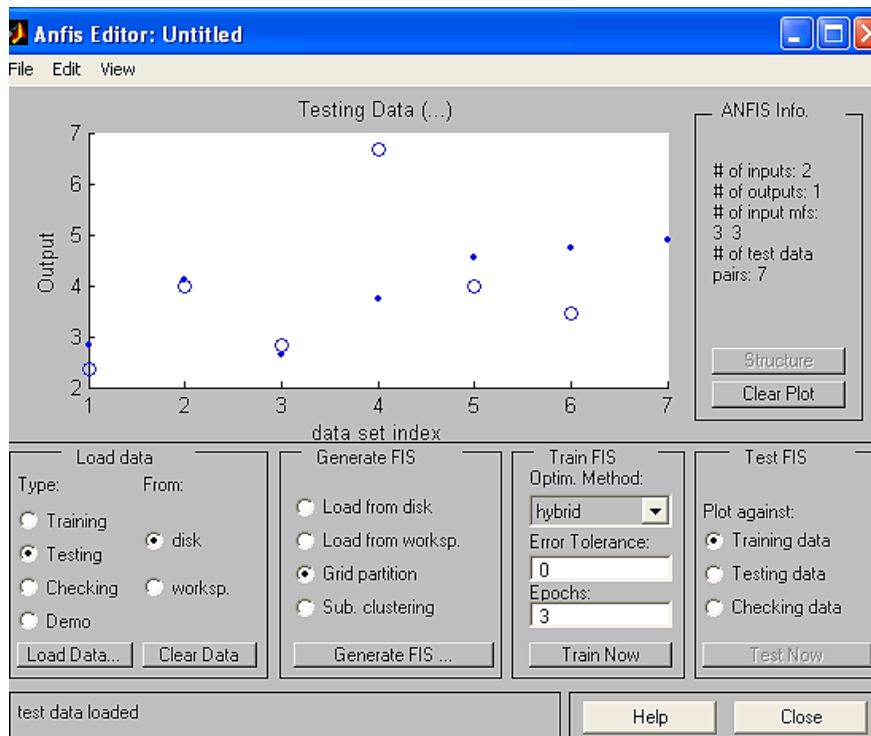


Рис. 10. Графічний інтерфейс редактора ANFIS після завантаження файлу *function.dat* з навчальними даними

Для знову створюваної гібридної мережі це вікно не містить ніяких даних. Для створення гібридної мережі необхідно завантажити дані. Для цієї мети слід скористатися кнопкою Load Data в лівій нижній частині графічного вікна. Водночас дані можуть бути завантажені із зовнішнього файлу

(disk) чи з робочої області (worksp). У першому випадку необхідно попередньо створити файл із вхідними даними (файл train), що являє собою звичайний текстовий файл. Після завантаження файлу із навчальними даними в редактор ANFIS у робочому вікні редактора буде зображений графік, ордината якого відображає значення вхідної математичної функції (кількість продукції в прикладі).

Завантажимо ще і тестові дані для перевірки. Для цього в лівому нижньому кутку екрана оберемо тип testing, натиснемо кнопку Load data і завантажимо файл test.txt.

*Крок 3.* Розпочати генерацію структури системи нечіткого виведення FIS. Ця структура аналогічна структурі системи нечіткого виведення типу Сугено. Тобто, має в своєму складі лінгвістичні вхідні змінні; вихідну змінну, терми якої представляються у вигляді числових констант або лінійних функцій від вхідних змінних; систему правил виведення.

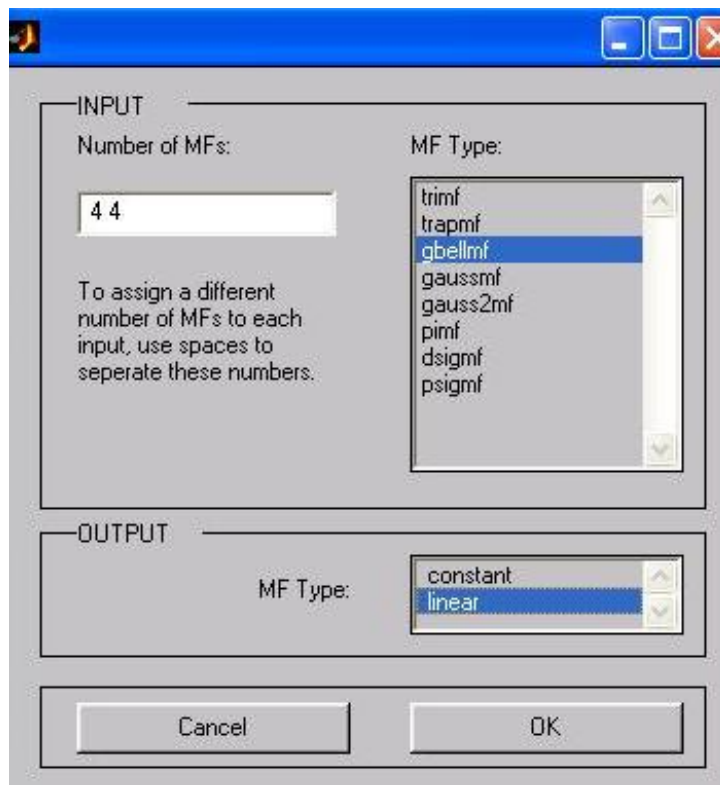
Крім того, можна завантажити структуру уже створеної FIS з диска (Load from disk), або з робочої області (Load from worksp). Під час створення структури нової FIS можна незалежно розбити усі вхідні змінні на області їх значень (Grid partision) чи скористатися процедурою субтрактивної кластеризації для попередньої розбивки значень вхідних змінних на кластери близьких значень (Sub. clustering).

Після натискання кнопки Generate FIS викликається діалогове вікно з вказівкою числа і типу функцій приналежності для окремих термів вхідних змінних і вихідної змінної (рис. 11). У цьому вікні можна вибрати тип функцій приналежності і кількість термів кожної вхідної змінної, а також тип вихідної змінної. Установивши параметри генерації, наприклад, як на рис. 11, одержимо структуру FIS.

Після генерації структури гібридної мережі можна її подивитись, натиснувши кнопку Structure в правій частині графічного вікна. Структура отриманої в результаті системи нечіткого виводу FIS відображається в окремому вікні і досить тривіальна за своїм видом.

*Крок 4.* Перед навчанням гібридної мережі необхідно задати параметри навчання, для чого варто скористатися наступною групою опцій у правій нижній частині робочого вікна:

1. Вибрати метод навчання гібридної мережі – зворотного розповсюдження (backpropo) чи гібридний (hybrid), що являє собою комбінацію методу найменших квадратів і методу спадання зворотного градієнта.



**Рис. 11. Діалогове вікно для завдання кількості і типу функцій приналежності**

2. Установити рівень помилки навчання (Error Tolerance) – за замовчуванням значення 0 (змінювати не рекомендується).

3. Задати кількість циклів навчання (Epochs) – за замовчуванням значення 3 (рекомендується збільшити і для розглянутого приклада задати його значення, що дорівнює 40).

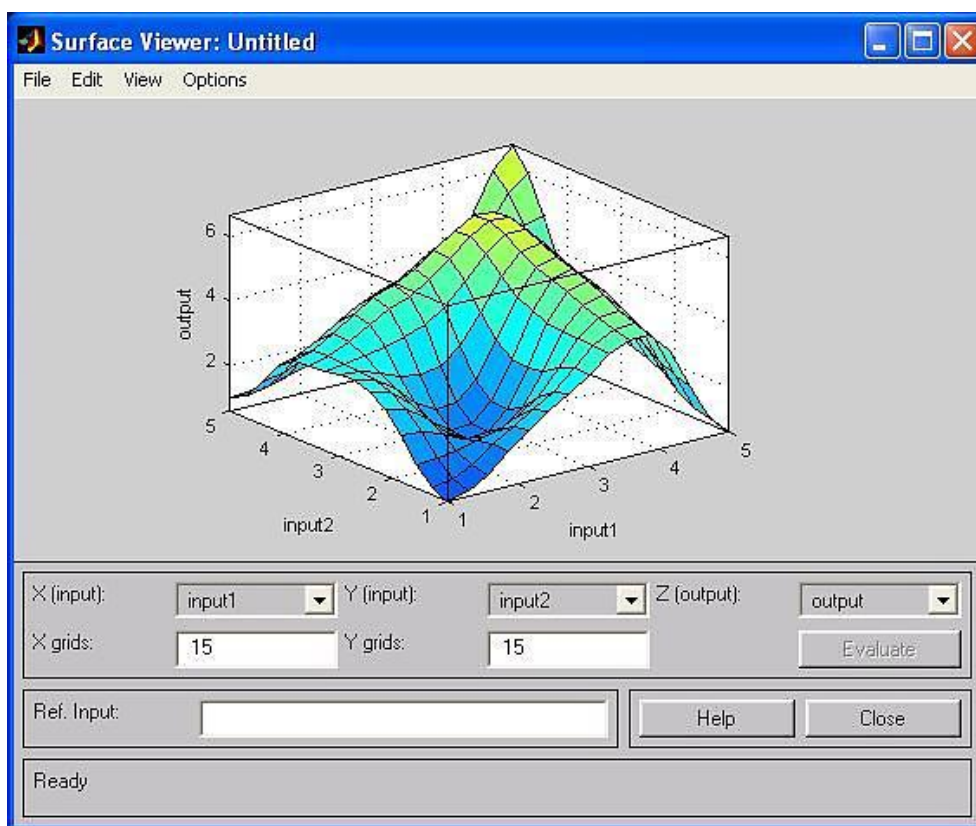
6. Для навчання мережі варто натиснути кнопку Train Now. Водночас хід процесу навчання ілюструється у вікні візуалізації у формі графіка – залежність помилки від кількості циклів навчання.

7. Аналогічно можуть бути виконані додаткові етапи тестування й перевірки гібридної мережі, для яких необхідно попередньо завантажити відповідні дані.

*Крок 5.* Подальше настроювання параметрів побудованої і навченої гібридної мережі може бути виконане за допомогою розглянутих раніше стандартних графічних засобів пакета Fuzzy Logic Toolbox. Для цього рекомендується зберегти створену систему нечіткого виведення в зовнішньому файлі з розширенням fis, після чого варто завантажити цей файл у редактор систем нечіткого виводу FIS. Можна працювати зі структурою системи нечіткого виводу безпосередньо в редакторі ANFIS.

Водночас також стають доступними редактор функцій приналежності системи нечіткого виводу (Membership Function Editor), редактор правил системи нечіткого виводу (Rule Editor), програма перегляду правил системи нечіткого виводу (Rule Viewer) і програма перегляду поверхні системи нечіткого виводу (Surface Viewer). Проаналізуйте правила та функції приналежності, які були побудовані даною системою апроксимації виробничої функції. Спробуйте їх корегувати і слідкувати за зміною результатів.

*Крок 6.* Виконати аналіз точності побудованої нечіткої моделі гібридної мережі можна за допомогою перегляду поверхні відповідної системи нечіткого виводу. Для цього слід записати побудовану мережу у вигляді файлу з розширенням \*.fis і завантажити його в системі нечіткого виведення fuzzyToolbox (рис. 12).

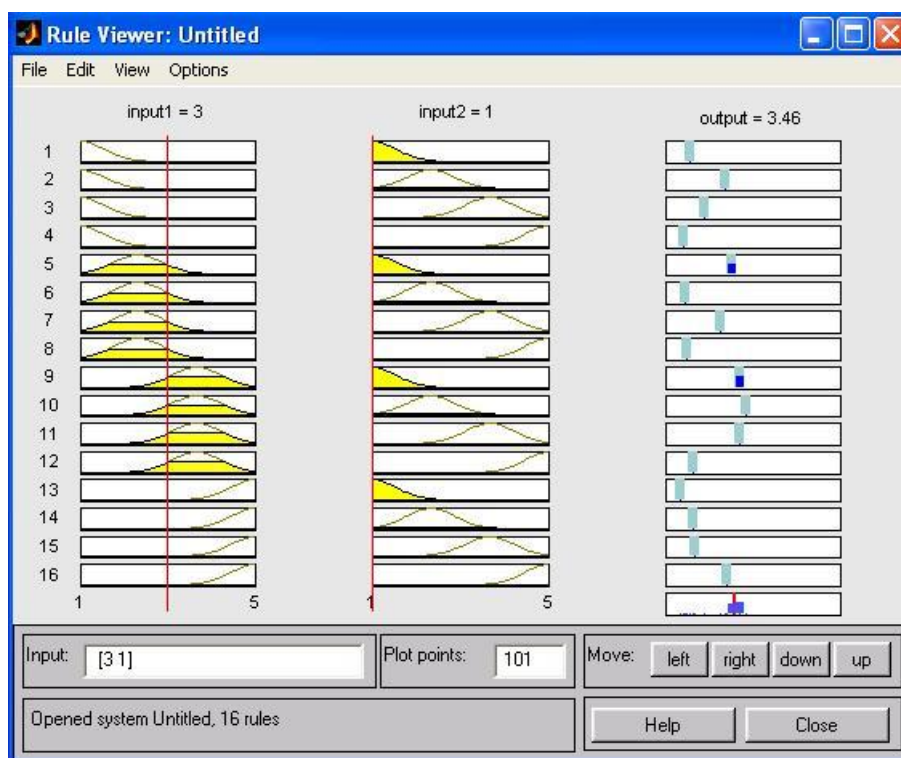


**Рис. 12. Графічний інтерфейс перегляду поверхні згенерованої системи нечіткого виведення**

Візуальний аналіз зображеного графіка з точним графіком функції  $q = F(x_1, x_2) = 2\sqrt{x_1}4\sqrt{x_2}$  дозволяє судити про досить високий ступінь їх співпадіння, що може свідчити про адекватність побудованої нечіткої моделі гібридної мережі.

Аналіз адекватності побудованої моделі можна виконати за допомогою перегляду правил відповідної системи нечіткого виведення (рис. 13).

Перевірка побудованої моделі гібридної мережі може бути виконана для декількох значень вихідної змінної. Із цією метою необхідно ввести конкретне значення в поле вводу Input (наприклад, значення [3 1]), після натискання клавіші <Enter> за допомогою побудованої моделі буде отримане відповідне значення вихідної змінної (у даному випадку значення 3.46). Порівнюючи отримане значення з точним значенням функції 3.46, одержимо відносну помилку порядку 0 %.



**Рис. 13. Графічний інтерфейс перегляду правил згенерованої системи нечіткого виведення**

Менш удаюю виявляється перевірка для значення вхідних змінних [4 3], для якого побудована модель пропонує значення 5.45 (табличне значення 5.26). Очевидно, що даний факт свідчить не на користь адекватності побудованої нечіткої моделі і вимагає її додаткового настроювання.

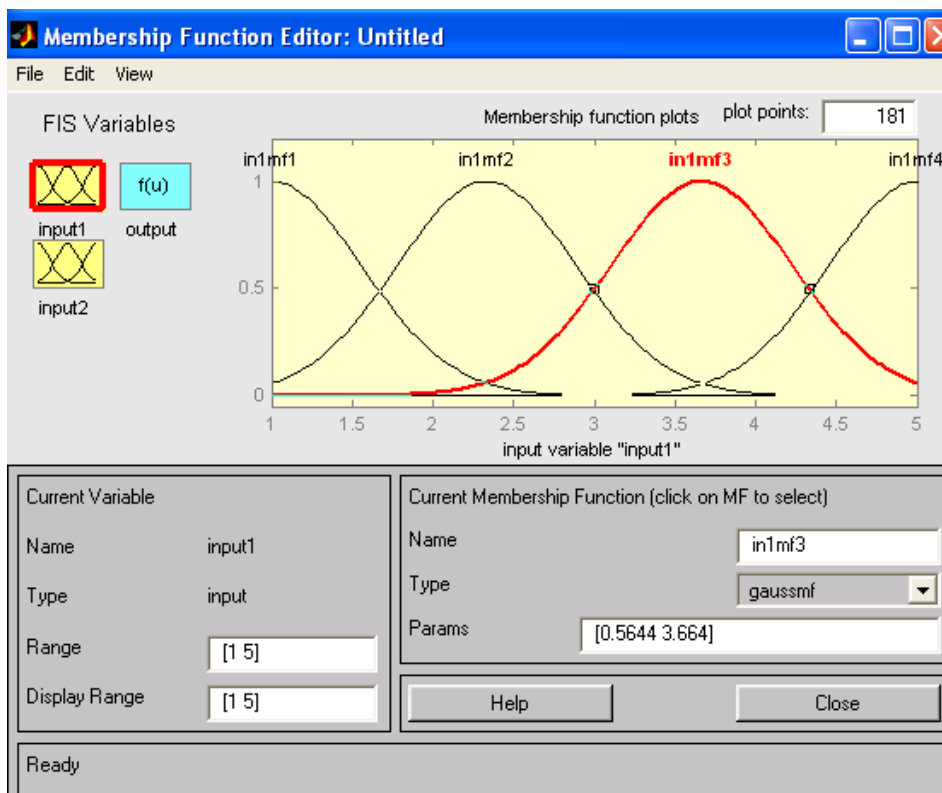
У загальному випадку додаткове настроювання моделі може бути виконане декількома можливими способами. Найбільш прийнятними з них є наступні:

- 1) підготовка і завантаження більшого за обсягом вибірки файлу з навчальними вхідними даними;

2) підготовка і завантаження додаткового файлу з перевірочними вхідними даними, сформованими для пар значень розглянутої математичної функції, відсутніх у вибірці навчальних даних;

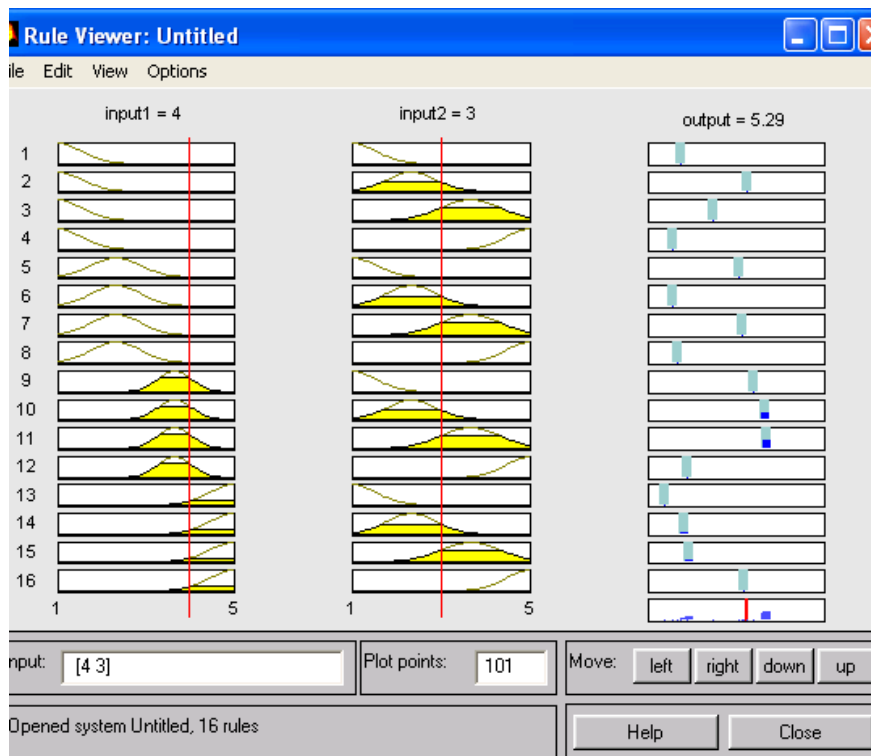
3) редагування типів і значень параметрів функцій приналежності термів вхідної і вихідної змінних за допомогою редактора функцій приналежності системи MATLAB.

Проілюструємо третій спосіб додаткового настроювання побудованої нечіткої моделі гібридної мережі. На перший погляд він є найбільш природним з погляду можливості візуального контролю виконуваних змін параметрів. Із цією метою відкриємо редактор функцій приналежності і методом підбора змінимо кількісні значення параметрів другої і третьої функції приналежності вхідної змінної, оскільки саме вони "працюють" під час одержання некоректного значення вихідної змінної: 5.45 для значення вхідних змінних 4 і 3 (рис. 14).



**Рис. 14. Графічний інтерфейс редактора функцій приналежності побудованої системи нечіткого виведення**

Після зміни параметрів у полі Params для третьої функції приналежності першої змінної на значення [0.385 3.66] одержимо практично точне значення вихідної змінної для значення вхідної змінної 5.29 (рис. 15).



**Рис. 15. Результат ручного настроювання параметрів функції приналежності вхідної змінної для розглянутої системи нечіткого виведення**

Більш ефективним способом настроювання, а точніше – модифікації, даної нечіткої моделі виявляється перший. Із цією метою можна збільшити обсяг навчальної вибірки до 12 пар значень, додавши ще три рядка у файл train.txt. Наприклад:

|   |   |      |
|---|---|------|
| 2 | 1 | 2.82 |
| 3 | 2 | 4.12 |
| 1 | 3 | 2.63 |

Після видалення усіх раніше завантажених даних кнопкою "Clear Data" завантажимо новий файл із навчальною вибіркою. Під час генерації структури нової FIS збільшимо кількість термів і, відповідно, кількість функцій приналежності вхідної змінної до 5, залишивши їх тип без зміни (gbellmf). Процес навчання виконаємо аналогічно раніше розглянутому. У результаті буде отримана нова система нечіткого виведення FIS, аналіз якої показує, що порівняно з першим варіантом нечіткої моделі вона більш точно описує вихідну математичну функцію. Це може бути основою для генерації нової нечіткої моделі, що пропонується виконати самостійно як вправу.

На закінчення необхідно відзначити, що навіть найпростіші розглянуті приклади відбивають творчий характер процесу побудови й аналізу моделей гібридних мереж. Водночас вибір того чи іншого способу додаткового настроювання нечітких моделей залежить не тільки від специфіки розв'язуваної задачі, але і від обсягу доступної вибірки навчальних і перевірочних даних.

У випадку недостатньої інформації навчальних даних використання гібридних мереж може виявитися взагалі недоцільним, оскільки одержати адекватну нечітку модель, а значить – і точний прогноз значень вихідної змінної не представляється можливим. Саме із цих причин необхідний попередній аналіз усіх можливостей застосовуваних нечітких моделей для рішення конкретних задач у тій чи іншій проблемній області. Подібний аналіз необхідно виконувати із системної точки зору і з урахуванням всіх обставин, що складаються на даний момент. Тільки всебічне і повне оцінювання проблемної ситуації дозволить розробити адекватну модель розв'язку тієї чи іншої конкретної задачі нечіткого керування чи ухвалення рішень.

## **Контрольні запитання для самодіагностики за модулями**

### **Змістовий модуль 1. Методи нечіткої логіки в аналізі статистичної інформації**

1. Невизначеність і неточність.
2. Ймовірнісний та нечіткий підхід к моделюванню невизначеності.
3. Стохастична невизначеність.
4. Лінгвістична невизначеність.
5. Лінгвістична невизначеність та нечіткість.
6. Означення нечіткої множини.
7. Основні поняття теорії нечітких множин.
8. Нечітка логіка.
9. Нечітка множина.
10. Нечітка величина.
11. Нечітка число.
12. Нечітка змінна.



13. Лінгвістична змінна.
14. Трапецеїдальні функції приналежності.
15. Прямі методи завдання функцій приналежності (фазифікації).
16. Посередні методи фазифікації.
17. Рекомендації з побудови функції приналежності.
18. Рівність і домінування нечітких множин.
19. Операція перетину нечітких множин.
20. Операція об'єднання нечітких множин.
21. Операція різниці нечітких множин.
22. Операція доповнення нечітких множин
23. Властивості операцій над нечіткими множинами.
24. Альтернативні операції перетину, об'єднання і різниці нечітких множин.
25. Нечіткі оператори.
26. Скалярна потужність дискретної нечіткої множини.
27. Порівняння кінцевих нечітких множин.
28. Арифметичні операції над нечіткими величинами.
29. Методи дефазифікації нечітких величин.
30. Метод центру тяжіння.
31. Метод центру площини.
32. Метод середини  $\alpha$ -зрізу.
33. Поняття нечіткого висловлення.
34. Прости і складні нечіткі висловлення.
35. Нечіткі логічні операції над нечіткими висловленнями.
36. Нечітке логічне заперечення.
37. Нечітка кон'юнкція.
38. Нечітка диз'юнкція
39. Нечітка імплікація.
40. Нечітка еквівалентність.
41. Нечіткі предикати.
42. Нечітке відношення і способи його завдання.
43. Основні характеристики нечітких відношень.
44. Операції над нечіткими відношеннями. Композиція бінарних нечітких відношень.
45. Принцип узагальнення.
46. Властивості бінарних нечітких відношень, які задані на одному універсумі.

47. Нечіткі графи.
48. Показник розмитості нечіткої величини.
49. Базова архітектура систем нечіткого висновку.
50. Основні етапи нечіткого висновку.
51. Застосування систем нечіткого висновку в задачах управління.
52. Моделі нечіткого висновку "ситуація – дія".
53. Моделі нечіткого висновку "ситуація – стратегія управління – дія".
54. Компетентність як нечітка величина.
55. Нечітка оцінка компетентності.
56. Базовий формалізм класичних мереж Петрі.
57. Розмітка мережі Петрі.
58. Властивості мереж Петрі.
59. Моделювання систем за допомогою мереж Петрі.
60. Нечіткі мережі Петрі.
61. Властивості нечітких мереж Петрі.
62. Класифікація нечітких мереж Петрі.
63. Застосування нечітких мереж Петрі в задачах у системах нечіткого висновку.
64. Загальна характеристика адаптивних систем нейро-нечіткого висновку.
65. Методика оцінювання якості об'єкта дослідження на основі лінгвістичних змінних.
66. Завдання лінгвістичних змінних "Якість об'єкта дослідження" та "Частковий показник якості об'єкта дослідження".
67. Знаходження загального показника якості об'єкта дослідження.
68. Нечіткі системи підтримки прийняття рішень.
69. Нечітке моделювання.
70. Нечітке прогнозування
71. Штучний інтелект.
72. Знання та моделі їх представлення.

## **Змістовий модуль 2. Використання нейронних мереж для дослідження соціально-економічних явищ**

1. Поняття: нейрон, нейромережа, нейрокомп'ютер, нейроінформатика.
2. Класифікація та види моделей нейромереж.
3. Властивості штучних нейромереж.

4. Загальне уявлення про навчання нейромереж.
5. Загальна характеристика та принципи побудови нейромереж.
6. Характеристики процесу навчання нейромереж.
7. Лінійна роздільність і лінійна нероздільність класів.
8. Порівняння моделей та алгоритмів навчання нейромереж прямого поширення.
9. Нейронні мережі у пакеті MATLAB. Модуль Neural Network Toolbox.
10. Пакет Statistica Neural Networks.
11. Моделі нейроелементів у пакеті MATLAB.
12. Біологічні нейрони. Математичні моделі нейроелементів.
13. Поняття: формальний нейрон, синапс, вага (ваговий коефіцієнт), поріг, дискримінантна (вагова) функція, функція активації, бажаний і реальний вихід нейромережі, навчання нейромережі, класифікація, апроксимація, оцінювання, помилка навчання / класифікації, час навчання / класифікації, цільова функція навчання.
14. Метод найменших квадратів як основа алгоритму Уїдроу – Хоффа. Чи завжди збігаються алгоритми навчання одношарового перцептрона?
15. Можливості і властивості одношарових перцептронів. Які задачі можна вирішувати на основі одношарових перцептронів, а які не можна? Чи доцільно застосовувати одношаровий перцептрон для класифікації складно (нелінійно) роздільних образів? Обґрунтуйте і доведіть відповідь. Наведіть приклади.
16. Чи впливає величина кроку навчання на час навчання одношарового перцептрона, багатшарової мережі? Відповідь обґрунтуйте теоретично та доведіть експериментально.
17. Які функції активації нейронів найчастіше використовують і чому? Чи впливає вид функції активації нейрона на тривалість навчання і роботи перцептрона, величину помилки навчання й класифікації (оцінювання) і, якщо впливає, то яким чином?
18. Чи впливає кількість використаних ознак на швидкість навчання перцептрона? Відповідь обґрунтуйте теоретично та доведіть експериментально.
19. У чому полягає подібність і відмінність біологічних і формальних нейронів?
20. Чи можна навчити дискретний одношаровий перцептрон обчислювати значення функцій:  $y = (x_1 \text{ and } x_2)$ ,  $y = (x_1 \text{ xor } x_2)$ ,  $y = \text{not} ((\text{not } x_1) \text{ and } x_2)$ , а дійсний одношаровий перцептрон – обчислювати значення функцій:  $y = 3x_1 - (-0,05)x_2$ ,  $y = \sin(x_1) + 0,3x_2$ ,  $y = 0,5x_1 + 2x_2 - 2,5(x_1 - x_2) + 5,5$ ? Відповіді обґрунтуйте і, якщо можливо, поясніть рисунками.

21. Багатошаровий перцептрон: модель і принципи побудови архітектури.

22. Які задачі можна розв'язувати на основі багатошарових перцептронів, а які не можна? Обґрунтуйте і доведіть відповідь. Наведіть приклади.

23. Алгоритм зворотного поширення помилки. Чи завжди збігається алгоритм зворотного поширення помилки для багатошарового перцептрона?

24. Які методи навчання багатошарових нейромереж вам відомі? Дайте порівняльну характеристику відомих вам методів навчання багатошарових нейромереж.

25. Градієнтні алгоритми навчання багатошарових нейромереж. Який із градієнтних методів навчання на основі зворотного поширення помилки ви рекомендували б для застосування, якщо критерієм оптимальності навчання нейронної мережі є: а) швидкість навчання; б) простота обчислень; в) універсальність?

26. Чи може дійсний багатошаровий перцептрон моделювати функцію  $y = x_1 + 0.5x_1 + x_2$ , якщо число його шарів дорівнює: а) 1; б) 2; в) 3? Відповіді обґрунтуйте і, якщо можливо, поясніть рисунками.

27. Чи можливе використання неградієнтних методів багатовимірної безумовної оптимізації для настроювання ваг багатошарових нейромереж, і, якщо можливе, то наскільки це доцільно робити?

28. Чи можна навчити дискретний багатошаровий перцептрон обчислювати значення функцій: 1)  $y = x_1$  and  $x_2$ , 2)  $y = x_1 \text{ xor } x_2$ , 3)  $y = \text{not} ((\text{not } x_1) \text{ and } x_2)$ , а дійсний багатошаровий перцептрон – обчислювати значення функцій: 1)  $y = 3x_1 - 0,05x_2$ ; 2)  $y = \sin(x_1) + 0,3x_2$ ; 3)  $y = 0,5x_1 + 2x_2 - 2,5(x_1 - (-x_2)) + 5,5$ ; 4)  $y = x_1 / (x_2 \sin(\pi))$ ? Відповіді обґрунтуйте і, якщо можливо, поясніть рисунками

29. Опишіть суть узагальненого градієнтного алгоритму. Окремі випадки узагальненого градієнтного алгоритму.

30. Нейронні мережі прямого поширення та градієнтні алгоритми навчання у пакеті MATLAB.

31. Моделі та принципи синтезу архітектури радіально-базисних нейромереж.

32. У чому подібність і відмінність радіально-базисних нейромереж і багатошарових нейромереж?

33. Застосування кластер-аналізу при навчанні радіально-базисних нейромереж.

34. Методи навчання радіально-базисних нейромереж.

35. Чи завжди збігаються алгоритми навчання радіально-базисних нейромереж?

36. Які задачі можна вирішувати на основі радіально-базисних нейромереж, а які не можна? Обґрунтуйте і доведіть відповідь. Наведіть приклади.

37. Радіально-базисні нейромережі у пакеті MATLAB.

38. Чи доцільно застосовувати радіально-базисних нейромереж для класифікації складно (нелінійно) роздільних образів?

39. Чи можна навчити радіально-базисну нейромережу обчислювати значення функцій: 1)  $y = x_1 \text{ and } x_2$ ; 2)  $y = x_1 \text{ xor } x_2$ ; 3)  $y = \text{not}((\text{not } x_1) \text{ and } x_2)$ ; 4)  $y = 3x_1 - 0,05x_2$ ; 5)  $y = \sin(x_1) + 0,3x_2$ ; 6)  $y = 0,5x_1 + 2x_2 - 2,5(x_1 - x_2) + 5,5$ ; 7)  $y = x_1 / (x_2 \sin(\pi))$ ? Відповіді обґрунтуйте і, якщо можливо, поясніть рисунками.

40. Переваги і недоліки радіально-базисних мереж.

41. Що таке генеральна сукупність, вибірка, екземпляр, ознака?

42. Вимоги до навчальних вибірок даних.

43. Що таке репрезентативна вибірка даних?

44. Чи повинна навчальна вибірка бути репрезентативною?

45. Чи повинна тестова вибірка бути репрезентативною?

46. Чи впливає обсяг навчальної вибірки на швидкість навчання нейромереж?

47. Чи впливає репрезентативність навчальної вибірки на точність класифікації екземплярів тестової вибірки?

48. Чи впливає репрезентативність тестової вибірки на точність класифікації екземплярів тестової вибірки?

49. Чи впливає репрезентативність тестової вибірки на точність навчання перцептрона по навчальній вибірці?

50. Чи залежить якість навчання нейромереж від якості та обсягу навчальної вибірки?

### **Змістовий модуль 3. Генетичні алгоритми в економіці**

1. Під час вирішення яких завдань комбінаторної оптимізації може бути використаний простий генетичний алгоритм із двійковим кодуванням хромосом?

2. Які модифікації необхідні для ефективного використання простого генетичного алгоритму для вирішення завдання укладання рюкзака?

3. Які види штрафних функцій можуть бути використані у фітнес-функції під час вирішення завдання укладання рюкзака?

4. Виконайте програмну реалізацію простого генетичного алгоритму на одній із мов програмування для розв'язання задачі укладання рюкзака з введенням у фітнес-функцію штрафної функції. Досліджуйте ефективність генетичного алгоритму в залежності від виду штрафної функції.

5. У чому полягає сутність алгоритму відновлення при рішення задачі укладання рюкзака?

6. Виконайте програмну реалізацію простого генетичного алгоритму на одній із мов програмування для розв'язання задачі укладання рюкзака з використанням алгоритму відновлення.

7. Виконайте програмну реалізацію простого генетичного алгоритму (ГА) на одній із мов програмування для розв'язання задачі укладання рюкзака з використанням алгоритму декодування.

8. Як може бути використаний простий ГА з двійковим кодуванням хромосом для вирішення завдання про покриття?

9. Чому неефективно двійкове кодування хромосоми під час вирішення завдання комівояжера?

10. Опишіть основні види недвійкового уявлення хромосоми для завдання комівояжера.

11. Опишіть "уявлення сусідства" і проблемно-орієнтовані оператори кросинговеру: обмін ребер, обмін турів, евристичний кросинговер.

12. Як може бути виконаний оператор мутації на поданні сусідства?

13. Опишіть "упорядковане представлення" і вкажіть який тип оператора кросинговеру може на ньому використовуватися.

14. Опишіть "уявлення шляхів" і проблемно-орієнтовані оператори кросинговеру: частково відповідної ОК (PMX), упорядкований ОК (OX), циклічний ОК (CX).

15. Які виконавчі матриці можна використовувати для подання туру?

16. Опишіть відповідні оператори кросинговеру для матриці суміжності.

17. Чим відрізняється матриця передування від матриці суміжності і яким чином можна реалізувати оператори кросинговеру на ній?

18. Придумайте свій спосіб кодування (подання) повного туру для завдання комівояжера і відповідні генетичні оператори.

19. Які методи застосовуються для генерації початкової популяції?

20. Які недоліки має "метод рулетки"?

21. Чим відрізняється ранжування від пропорційного відбору?

22. Що таке локальний відбір?
23. Опишіть метод турнірного відбору.
24. Як використовується метод Больцмана під час відбору особин?
25. Опишіть методи відбору пар для схрещування.
26. Що таке неявні методи відбору?
27. Опишіть двійкову рекомбінацію.
28. Чим відрізняється багатоточковий кросинговер від класичного?
29. Що таке однорідний кросинговер?
30. Чим відрізняється рекомбінація дійсних чисел від класичного кросинговеру?
31. Що таке дискретна рекомбінація?
32. Опишіть проміжну рекомбінацію.
33. Чим відрізняється лінійна рекомбінація від проміжної?
34. Що таке інверсія?
35. Як виконується мутація над числами?
36. Чим відрізняється неоднорідна мутація від звичайної?
37. Які існують методи скорочення популяції?
38. В яких випадках доцільно застосовувати генетичний мікроалгоритм?
39. Опишіть нестаціонарний генетичний алгоритм.
40. Чим замінюється відбір батьків в нестаціонарному генетичному алгоритмі?
41. Які методи визначення термінів життя ви знаєте?
42. Що таке ніші в генетичному алгоритмі?
43. Чим еволюція Ламарка відрізняється від еволюції Дарвіна?
44. Опишіть гібридний генетичний алгоритм на основі еволюції Ламарка.
45. У чому полягає адаптація в генетичному алгоритмі?
46. Як змінюються ймовірності кросинговеру і мутації при адаптації?
47. Які види адаптації генетичних алгоритмів ви знаєте?
48. Як можна виконати адаптацію числа особин популяції?
49. Як можна виконати адаптацію значень ймовірностей кросинговеру і мутації.

## Рекомендована література

### Основна

1. Алтунин А. Е. Модели и алгоритмы принятия решений в нечетких условиях / А. Е. Алтунин, М. В. Семухин. – Тюмень : Изд-во Тюменского государственного университета, 2000. – 352 с.
2. Борисов А. Н. Модели принятия решений на основе лингвистической переменной / А. Н. Борисов, А. В. Алексеев, О. А. Крумберг. – Рига : Зинатне, 1982. – 256 с.
3. Борисов А. Н. Обработка нечеткой информации в системах принятия решений / А. Н. Борисов, А. В. Алексеев, Г. В. Меркурьева. – Москва : Радио и связь, 1989. – 304 с.
4. Борисов А. Н. Принятие решений на основе нечетких моделей. Примеры использования / А. Н. Борисов, О. А. Крумберг, И. П. Федоров. – Рига : Зинатне, 1990. – 184 с.
5. Бочарников В. П. Fuzzy-технология: математические основы практика моделирования в экономике / В. П. Бочарников. – Санкт-Петербург : Наука РАН , 2001. – 328 с.
6. Дюбуа Д. Теория возможностей. Приложения к представлению знаний в информатике / Д. Дюбуа, А. Прад. – Москва : Радио и связь, 1990. – 288 с.
7. Заде Л. А. Основы нового подхода к анализу сложных систем и процессов принятия решений. В кн.: Математика сегодня / Л. А. Заде. – Москва : Знание, 1974. – С. 5 – 49.
8. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л. А. Заде. – Москва : Мир, 1976. – 165 с.
9. Зайченко Ю.П. Исследование операций : нечеткая оптимизация : учеб. пособ. / Ю. П. Зайченко. – Киев: Высшая школа, 1991. – 191 с.
10. Кофман А. Введение в теорию нечетких множеств / А. Кофман – Москва: Радио и связь, 1982. – 432 с.
11. Недосекин А. О. Нечетко-множественный анализ риска фондовых инвестиций / А. О. Недосекин. – Санкт-Петербург : Изд-во "Сезам", 2002. – 181 с.



12. Нечеткие множества в моделях управления и искусственного интеллекта / А. Н. Аверкин, И. З. Батыршин, А. Ф. Блишун и др. / под ред. Д. А. Поспелова. – Москва : Наука. гл. ред. физ.-мат. лит., 1986. – 312 с.
13. Нечеткие множества и теория возможностей. Последние достижения / под ред. Р. Р. Ягера. – Москва : Радио и связь, 1986. – 408 с.
14. Орловский С. А. Проблемы принятия решений при нечеткой информации / С. А. Орловский. – Москва : Наука, 1981. – 206 с.
15. Поспелов Д. А. Логико-лингвистические модели в системах управления / Д. А. Поспелов. – Москва : Энергоиздат, 1981. – 232 с.
16. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети / А. П. Ротштейн. – Винница : УНИВЕРСУМ-Винница, 1999. – 320 с.
17. Ротштейн А. П. Нечеткая надежность алгоритмических процессов / А. П. Ротштейн, С. Д. Штовба. – Винница : Континент-ПРИМ, 1997. – 142 с.
18. Ротштейн А. П. Проектирование нечетких баз знаний: лабораторный практикум и курсовое проектирование : учеб. пособ / А. П. Ротштейн, С. Д. Штовба. – Винница: Винницкий государственный технический университет, 1999. – 65 с.
19. Усков А. А. Интеллектуальные системы управления на основе методов нечеткой логики / А. А. Усков, В. В. Круглов. – Смоленск : Смоленская городская типография, 2003. – 177 с.
20. Усков А. А. Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика / А. А. Усков, А. В. Кузьмин. – Москва : Горячая линия – Телеком, 2004. – 144 с.
21. Ярушкина Н. Г. Основы теории нечетких и гибридных систем : учеб. пособ / Н. Г. Ярушкина. – Москва : Финансы и статистика, 2004. – 320 с.

### **Додаткова**

22. Алиев Р. А. Производственные системы с искусственным интеллектом / Р. А. Алиев, Н. М. Абдикеев, М. М. Шахназаров. – Москва : Радио и связь, 1990. – 264 с.
23. Беллман Р. Принятие решений в расплывчатых условиях / Р. Беллман, Л. Заде. В кн.: Вопросы анализа и процедуры принятия решений. – Москва : Мир, 1976. – С. 172–215.

24. Берштейн Л. С. Нечеткие модели принятия решений: дедукция, индукция, аналогия / Л. С. Берштейн, А. В. Боженюк. – Таганрог : Изд-во ТРТУ, 2001. – 110 с.

25. Воцинин А. П. Оптимизация в условиях неопределенности / А. П. Воцинин, Г. Р. Сотиров. – Москва : Изд-во МЭИ (СССР) и Техника (НРБ), 1989. – 224 с.

26. Демидова Л. А. Алгоритмы и системы нечеткого вывода при решении задач диагностики городских инженерных коммуникаций в среде MATLAB / В. В. Кираковский, А. Н. Пылькин. – Москва : Радио и связь, Горячая линия – Телеком, 2005. – 365 с.

27. Леоненков А. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. Леоненков. – Санкт-Петербург : БХВ-Петербург – 2003. – 736 с.

28. Малышев Н. Г. Нечеткие модели для экспертных систем в САПР / Л. С. Бернштейн, А. В. Боженюк. – Москва : Энергоиздат, 1991. – 136 с.

29. Мелихов А. Н., Ситуационные советующие системы с нечеткой логикой / А. Н. Мелихов, Л. С. Берштейн, С. Я. Коровин. – Москва : Наука, 1990. – 272 с.

30. Минаев Ю. Н. Методы и алгоритмы решения задач идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе / Ю. Н. Минаев, О. Ю. Филимонова, Лиес Бенамеур. – Москва : Горячая линия – Телеком, 2003. – 205 с.

31. Митюшкин Ю. И. Soft-Computing: идентификация закономерностей нечеткими базами знаний / Ю. И. Митюшкин, Б. И. Мокин, А. П. Ротштейн. – Винница : УНІВЕРСУМ-Вінниця, 2002. – 145 с.

32. Орлов А. И. Задачи оптимизации и нечеткие переменные / А. И. Орлов. – Москва : Знание, 1980. – 64 с.

33. Системы фуцци-управления / В. И. Архангельский, И. Н. Богаенко, Г. Г. Грабовский, Н. А. Рюмшин. – Киев : Техника, 1997. – 208 с.

## **Інформаційні ресурси**

34. Центр компетенцій MathWorks. – Режим доступу : <https://matlab.ru>.

35. Проектування систем управління Fuzzy Logic Toolbox. Режим доступу : <http://matlab.exponenta.ru/fuzzylogic>.

## Зміст

|  |    |
|--|----|
| Вступ.....   | 3  |
| Запитання для самодіагностики за модулями та темами.....   | 5  |
| Змістовий модуль 1. Методи нечіткої логіки в аналізі статистичної інформації .....                                       | 5  |
| Змістовий модуль 2. Використання нейронних мереж для дослідження соціально-економічних явищ .....                        | 6  |
| Змістовий модуль 3. Генетичні алгоритми в економіці .....  | 8  |
| Завдання для самостійної роботи за темою 3. "Проектування систем нечіткого висновку на основі алгоритму Мамдані" .....   | 8  |
| Завдання для самостійної роботи за темою 6. "Розроблення інтелектуальних систем на основі моделей нейронних мереж" ..... | 18 |
| Контрольні запитання для самодіагностики за модулями .....   | 32 |
| Змістовий модуль 1. Методи нечіткої логіки в аналізі статистичної інформації .....                                       | 32 |
| Змістовий модуль 2. Використання нейронних мереж для дослідження соціально-економічних явищ .....                        | 34 |
| Змістовий модуль 3. Генетичні алгоритми в економіці .....  | 37 |
| Рекомендована література.....  | 40 |
| Основна .....  | 40 |
| Додаткова .....  | 41 |
| Інформаційні ресурси .....   | 42 |

НАВЧАЛЬНЕ ВИДАННЯ

# ІНТЕЛЕКТУАЛЬНІ МЕТОДИ ПРОГНОЗУВАННЯ СОЦІАЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ

**Методичні рекомендації  
до самостійної роботи студентів  
спеціальності 051 "Економіка"  
другого (магістерського) рівня**

*Самостійне електронне текстове мережеве видання*

Укладачі: **Стрижиченко Костянтин Анатолійович**  
**Степуріна Світлана Олександрівна**

Відповідальний за видання *О. В. Раєвська*

Редактор *О. В. Анацька*

Коректор *О. В. Анацька*

План 2019 р. Поз. № 108 ЕВ. Обсяг 44 с.

---

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

---

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру  
ДК № 4853 від 20.02.2015 р.*