

Міністерство освіти і науки України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Мартинчук Тетяна Олегівна

**Алгоритми побудови нечітких продукційних правил
на основі аналізу біомедичних зображень /
Development algorithms of fuzzy production rules on
based biomedical images analysis**

Спеціальність 8.091501 – Комп'ютерні системи та мережі

Дипломна робота за освітньо-кваліфікаційним рівнем «магістр»

Науковий керівник
д.т.н., професор Березький О.М.

Дипломну роботу допущено до захисту

«__» _____ 20__ р.

Зав. кафедри КІ

Березький О.М. _____

Тернопіль – 2017

ЗМІСТ

Перелік умовних скорочень.....	7
Вступ.....	8
1 Аналіз методів, моделей та програмних засобів представлення знань для діагностування.....	11
1.1 Побудова тестової вибірки цитологічних та гістологічних зображень.....	11
1.2 Моделі представлення знань.....	17
1.3 Основні поняття нечіткої логіки	22
1.4 Аналіз програмних засобів представлення знань.....	28
1.5 Постановка завдання дипломної роботи.....	34
2 Алгоритми побудови нечітких продукційних правил.....	36
2.1 Нечітка модель представлення знань та її модифікації.....	36
2.2 Побудова функцій належності лінгвістичних змінних.....	41
2.3 Алгоритми фазифікації та дефазифікації.....	49
2.4 Узагальнений алгоритм побудови нечітких правил.....	52
3 Програмна реалізація нечітких продукційних правил.....	56
3.1 Програмно-апаратний засіб.....	56
3.2 Побудова функцій належності.....	65
3.3 Побудова нечітких продукційних правил.....	70
3.4 Аналіз нечітких продукційних правил.....	73
Висновки.....	77
Список використаних джерел.....	78
Додаток А Копія екрану вікна нечіткої системи.....	84
Додаток Б Вихідний текст нечіткої системи.....	85
Додаток В Копія публікації.....	99
Додаток Г Довідка про впровадження.....	108

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

РМЗ – рак молочної залози

БД– база даних

БЗ – база знань

СУБД – система управління базою даних

ЛЗ – лінгвістична змінна

НЧ – нечітке число

НМ – нечітка множина

ФН – функція належності

НПП – нечітке продукційне правило

ОС – операційна система

ПАК – програмно-апаратний комплекс

ПЗ – програмне забезпечення

FLT англ. – Fuzzu Logic Toolbox

FIS англ. – Fuzzy Inference System

ВСТУП

Актуальність. Розповсюдженість раку молочної залози посідає перше місце в структурі онкологічних захворювань жінок. У світі щорічно діагностується близько одного мільйона нових випадків даного захворювання, і кожен рік від нього помирає близько півмільйона жінок, вірогідність захворіти істотно зростає.

Рак молочної залози (РМЗ) в Україні, так само, як і в більшості країн світу, з кожним роком розповсюджується і нині посідає перше місце в структурі захворюваності і смертності від злоякісних новоутворень серед жінок (25% від усіх випадків раку). В Україні рак молочної залози є однією з основних медико-соціальних проблем сучасної системи охорони здоров'я внаслідок значної поширеності та високого рівня інвалідності та смертності від нього.

Важливо пам'ятати, що рак – це діагноз, і своєчасне виявлення і розпочате лікування дає реальний шанс на одужання. Друга причина подібної картини полягає в тому, що хоча патологія молочної залози виявляється у 90% жінок, вони проте не роблять необхідних кроків і звертаються до лікаря лише на пізніх стадіях хвороби. Третя причина високої смертності від раку молочної залози – це відсутність масових програм раннього виявлення і профілактики цього захворювання у жінок.

Експертна постановка діагнозу базується на якісних показниках аналізу біомедичних зображень, а оцінка передракових станів в основному базується на кількісних характеристиках ознак мікрооб'єктів. Поширеність програмних засобів опрацювання біомедичних зображень дозволяє зробити процес прийняття правильного діагнозу швидшим та точнішим для експерта.

Аналіз злоякісних процесів на основі цитологічних зображень і гістологічних зображень виконується візуально, що є необхідною, але недостатньою, оскільки для повної характеристики патологічних процесів потрібні кількісні ознаки, які отримуються шляхом числового

морфометричного аналізу, що зменшує суб'єктивізм дослідника. Для автоматизації аналізу гістологічних та цитологічних зображень в умовах клінічної практики застосовуються системи автоматизованої мікроскопії.

Мета роботи полягає в аналізі біомедичних зображень передракових станів і раку молочної залози та побудові нечітких продукційних правил діагностування. Щоб досягти поставленої мети необхідно здійснити такі завдання:

- 1) дослідити відомі моделі представлення знань;
- 2) визначити функції належності на основі кількісних ознак мікрооб'єктів отриманих від експерта;
- 3) побудувати нечіткі продукційні правила;
- 4) здійснити експериментальні дослідження передракових та ракових станів молочної залози.

Об'єкт дослідження – гістологічні та цитологічні зображення раку молочної залози.

Предмет дослідження – алгоритми побудови нечітких продукційних правил.

Методи дослідження базуються на використанні методів комп'ютерного зору, моделей представлення знань, теорії нечіткої логіки.

Наукова новизна полягає у розробці алгоритмів побудови нечітких продукційних правил на основі аналізу біомедичних зображень, що містить кількісні і якісні ознаки мікрооб'єктів.

Практичне значення полягає у побудові нечітких продукційних правил постановки діагнозу (передракових і ракових станів молочної залози) та проведенні експериментальних досліджень в середовищі Fuzzy Logic Toolbox.

Публікація та апробація ДР. Результати наукового дослідження опубліковано в матеріалах VI Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2016 [1].

Впровадження результатів дипломної роботи здійснено при виконанні

держбюджетної теми "КІ-05-2016" «Гібридна інтелектуальна інформаційна технологія діагностування передракових станів молочної залози на основі аналізу зображень».

У першому розділі проведено аналіз бази даних цитологічних та гістологічних зображень, досліджено відомі моделі представлення знань та програмні засоби представлення знань.

У другому розділі проведено аналіз нечітких моделей представлення знань, аналіз алгоритму фазифікації та дефазифікації та розроблено узагальнений алгоритм побудови нечітких правил.

У третьому розділі визначено функції належності на основі кількісних ознак мікрооб'єктів, отриманих від експерта, побудовано функції належності в середовищі Fuzzy Logic Toolbox, побудовано нечіткі продукційні правила в середовищі Fuzzy Logic Toolbox та здійснено експериментальні дослідження передракових та ракових станів молочної залози.

У додатках наведено копію екрану вікна нечіткої системи та вихідний текст нечіткої системи.

1 АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА ПРОГРАМНИХ ЗАСОБІВ

ПРЕДСТАВЛЕННЯ ЗНАНЬ ДЛЯ ДІАГНОСТУВАННЯ

1.1 Побудова тестової вибірки цитологічних та гістологічних зображень

Зображення відіграють важливу роль в процесі постановки діагнозу. Насправді більшу частину недавнього прогресу, зокрема, в діагностиці, можна простежити від наявності більш складних методів візуалізації, які не тільки показують структуру тіла в найдрібніших деталях, але також можуть показати функцію тканин в організмі.

Для верифікації попереднього діагнозу, подальшого лікування і планування терапії використовуються, отримані за допомогою мікроскопа, гістологічні та цитологічні зображення, які входять до класу біомедичних зображень.

Поняття біомедичних зображень охоплює собою науку і галузь медицини, пов'язаних з розвитком і використанням пристроїв і методів візуалізації для отримання внутрішніх анатомічних зображень, забезпечує біохімічний та фізіологічний аналіз тканин і органів.

Поле біомедичних зображень використовує передові технології для збору, зберігання, аналізу і відображення зображень органу, тканини, клітинному і молекулярному рівні. Умови такі, як комп'ютерна томографія, магнітно-резонансної томографії, молекулярні візуалізації, і сканування мікроскопії відіграють все більш важливу роль в клінічній практиці і фундаментальних дослідженнях науки про життя [2].

Гістологія (від грец. *histos* – тканина, *logos* – вчення) – наука про будову, розвитку і життєдіяльність тканин тварин організмів. Тканини являють собою систему клітин і неклітинних структур, що об'єдналися і спеціалізувалися в процесі еволюції для виконання найважливіших функцій в організмі. Для кожної з основних тканинних систем характерні притаманні саме їм

особливості будови, розвитку і життєдіяльності. Предметом загальної гістології, або власне вчення про тканинах, є загальні закономірності, притаманні тканинному рівнем організації та відмінні особливості конкретних тканин; предметом приватної гістології – закономірності життєдіяльності та взаємодії різних тканин в органах на більш високих рівнях організації. Приватна гістологія служить основою для вивчення мікроскопічної будови морфофункціональних одиниць органів та органів в цілому [3]. Гістологічні зображення – це мікроскопічні зображення препаратів тонких зрізів біологічної тканини, наведено на рисунку 1.1.

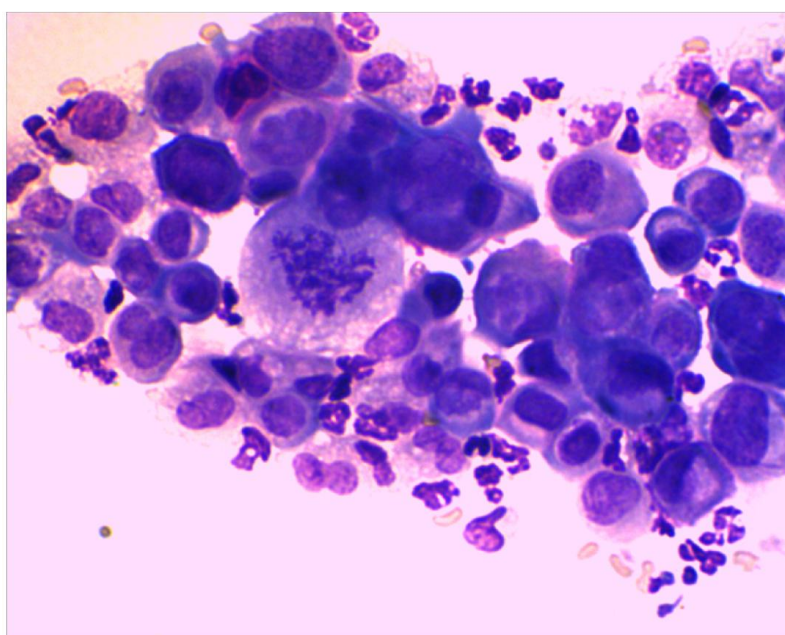


Рисунок 1.1 – Гістологічне зображення

Курс гістології включає в себе також цитологію – вчення про клітини. Цитологія (від грец. *kytos* – клітина, *logos* – вчення) – наука про розвиток, будову і життєдіяльності клітин. Цитологічні зображення – це мікроскопічні зображення препаратів, що містять клітини та їхні складові частини (ядро, вакуолі і та інші), показано на рисунку 1.2. Цитологія складає необхідну частину гістології, так як клітини є основою розвитку і будови тканин. Цитологія в останні роки збагатилася багатьма науковими відкриттями, які

зробили істотний внесок у розвиток біологічних і медичних наук і практику охорони здоров'я. Нові дані про структуру ядра, його хромосомного апарату лягли в основу цитодіагностики спадкових захворювань, пухлин, хворіб крові та багатьох інших хворіб. Розкриття особливостей ультраструктури та хімічного складу клітинних мембран є основою для розуміння закономірностей взаємодії клітин в тканинних системах, захисних реакціях і ін. У медичній практиці широко використовується цитодіагностика. Клітини здорового і хворого організму вивчаються в мазках крові та кісткового мозку, спинномозкової рідини, слини, сечі, у зразках різних органів, узятих при біопсії [4].

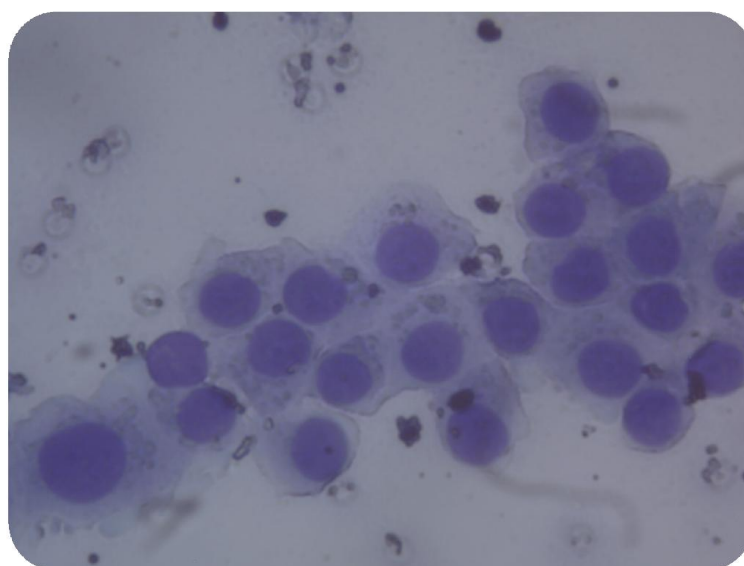


Рисунок 1.2 – Цитологічне зображення

У сучасній гістології та цитології застосовуються різноманітні методи дослідження, що дозволяють всебічно вивчати процеси розвитку, будови і функції клітин, тканин і органів. Основними етапами цитологічного і гістологічного аналізу є вибір об'єкта дослідження, підготовка його для вивчення в мікроскопі, застосування методів мікроскопіювання, а також якісний і кількісний аналіз зображень.

Діагностування ракових захворювань зазвичай вимагає цитологічного та гістологічного дослідження. Цитологічний метод діагностики захворювань

визначив себе як безпечний, простий, швидкий, не дорогий та достатньо інформативний, що дозволяє використовувати його для морфологічного встановлення діагнозу при діагностуванні захворювань молочної залози, органів грудної порожнини, лімфатичних вузлів, м'яких тканин. Цей метод дозволяє встановити характер ураження, гістогенез. Інформативність отриманого матеріалу дозволяє встановити діагноз, від'ємні результати не виключають наявності злоякісного процесу. Повторне отримання матеріалу дозволяє з більшим ступенем достовірності встановити діагноз. Проведення гістологічних та цитологічних перевірок – найбільш ефективний метод контролю якості цитологічних досліджень [5].

Об'єктами дослідження служать живі і мертві (фіксовані) клітини і тканини, і їх зображення, отримані в світлових і електронних мікроскопах. Основним об'єктом дослідження є гістологічні препарати, приготовані з фіксованих структур. Препарат може являти собою мазок (наприклад, мазок крові, кісткового мозку, слини, спинномозкової рідини та ін.), Відбиток (наприклад, селезінки, печінки), плівку з тканини (наприклад, сполучної або очеревини, плеври, м'якої мозкової оболонки), тонкий зріз. Найбільш часто для вивчення використовується зріз тканини або органу. Гістологічні препарати можуть вивчатися без спеціальної обробки. Наприклад, приготований мазок крові, відбиток, плівка або зріз органу можуть відразу розглядатися під мікроскопом. Але внаслідок того, що структури мають слабкий контраст, вони погано виявляються в звичайному світловому мікроскопі і потрібне використання спеціальних мікроскопів (фазово-контрастні та інші). Тому частіше застосовують спеціально оброблені препарати: фіксовані, укладені у тверде середовище і пофарбовані [6].

Аналіз злоякісних процесів на основі цитологічних зображень і гістологічних зображень виконується візуально, що є необхідною, але недостатньою, оскільки для повної характеристики патологічних процесів потрібні кількісні ознаки, які отримуються шляхом числового

морфометричного аналізу, що зменшує суб'єктивізм дослідника. Для автоматизації аналізу гістологічних та цитологічних зображень в умовах клінічної практики застосовуються системи автоматизованої мікроскопії.

Для кількісного опису мікрооб'єктів використовують такі ознаки: площа клітини, площа ядра клітини, площа цитоплазми клітини, периметр клітини, периметр ядра клітини, периметр цитоплазми клітини, геометричні ознаки форми і т.п. Для кількісного опису патологічних змін у структурах використовують кількість шарів клітин у тканині, коефіцієнт структурної атипії та інші [7].

Для опису мікрооб'єктів та їх якісних ознак потрібно формалізувати знання (досвід) експерта морфолога: логічні дерева рішень, мову опису клітини та діагнозу. Описові знання можна формалізувати у вигляді спеціальної онтології опису клітин, яка містить всю інформацію про морфологію ракових клітин. Структурована та ієрархічна онтологія складається з усіх значних морфологічних характеристик, які розглядаються окремо. Для кожної характеристики (наприклад, розмір ядра) необхідно зафіксувати якісні значення категорій які можна присвоїти цій ознаці (наприклад, дуже малий, малий, середній або великий) [8]. Морфологічні характеристики і значення повинні інтерпретуватися і «вимірюватися» інтуїтивно, так щоб онтологія застосовувалась і для автоматичного і для ручного морфологічних досліджень.

Для тестування та налаштування комп'ютерних систем опрацювання та класифікації зображень дослідники активно створюють і поширюють навчальні вибірки гістологічних і цитологічних зображень. Принциповою відмінністю такої бази даних є надання підтвердженого діагнозу, наведення виділених границь мікрооб'єктів та їх числових ознак.

База даних дозволяє створювати навчальну вибірку для випробування і перевірки систем аналізу гістологічних і цитологічних зображень.

Система баз даних – це, по суті, не що інше, як комп'ютеризована система зберігання однотипних записів. Саму ж базу даних можна розглядати як подобу

електронної картотеки, тобто сховище або контейнер для деякого набору файлів даних, занесених в комп'ютер. Користувачам цієї системи надається можливість виконувати (або передавати системі запити на виконання) багато різних операцій над такими файлами, наприклад:

- 1) додавати нові порожні файли в базу даних;
- 2) додавати нові дані в існуючі файли;
- 3) отримувати дані з існуючих файлів;
- 4) видаляти дані з існуючих файлів;
- 5) змінювати дані в існуючих файлах;
- 6) видаляти існуючі файли з бази даних.

До апаратного забезпечення для взаємодії бази даних необхідна система, відноситься наступне:

1) томи вторинній (зовнішній) пам'яті (зазвичай це магнітні диски), використовувані для зберігання інформації, а також відповідні пристрої введення-виведення (дискководи тощо), контролери пристроїв, канали вводу-виводу і т.д .;

2) апаратний процесор (або процесори) разом з оперативною (первинної) пам'яттю, призначені для підтримки роботи програмного забезпечення системи баз даних [9].

Між власне фізичною базою даних (тобто даними, які реально зберігаються на комп'ютері) і користувачами системи розташовується рівень програмного забезпечення, який можна називати по-різному: диспетчер бази даних (database manager), сервер бази даних (database server) або, що більш звично, система управління базами даних (database management system – DBMS). Всі запити користувачів на отримання доступу до бази даних (рисунок 1.3) обробляються СУБД. Для збереження інформації про дослідження в системі передбачено підключення до мережевої або локальної БД, що підтримує специфікацію системи відкритої бази даних.

COLUMN_N...	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DE...	EXTRA
id	int(11)	NO	PRI	(Null)	auto_incre...
colorationM...	varchar(50)	NO		(Null)	
comment	varchar(255)	NO		(Null)	
date	datetime	NO		(Null)	
glass	varchar(50)	NO	MUL	(Null)	

Рисунок 1.3 – Вікно перегляду структури та значень таблиць в базі даних

Система дозволяє сформувати такі звіти по: всій БД, пацієнту, дослідженню, скелету та зображенню. Звіт можна сформувати у таких форматах: Microsoft Excel 97/2000/XP/2003 (.xls), Microsoft Excel 2007/2010 (.xlsx), (WEKA) Attribute-Relation File Format (.arff), CSV data file (.csv). Для здійснення інтелектуального аналізу даних реалізовано функцію експорту бази даних у форматі «.arff» в програмну систему Weka [10].

1.2 Моделі представлення знань

Важливе місце в теорії систем штучного інтелекту займає проблема представлення знань, яка в загальному випадку може бути поділена на наступні класи та представлена на рисунку 1.4.

Перший підхід – емпіричним, заснований на вивченні принципів організації людської пам'яті і моделюванні механізмів вирішення завдань людиною. На основі цього підходу в даний час розроблено і отримали найбільшу популярність наступні моделі:

1) продукційні моделі – модель заснована на правилах, дозволяє представити знання у вигляді пропозицій типу: «ЯКЩО умова, ТО дія», де

умовами є твердження про вміст якоїсь бази даних, а дії являють собою процедури, які можуть змінювати вміст бази даних. Продукційна модель володіє тим недоліком, що при накопиченні досить великого числа (порядку декількох сотень) продукцій вони починають суперечити один одному. У продукційних моделях виділяють три основні компоненти:

- 1) неструктурована або структурована БД;
- 2) набір продукційних правил або продукцій;
- 3) інтерпретатор, який послідовно визначає, які продукції можуть бути активовані в залежності від умов, в них містяться; вибирає одне з застосовних в даній ситуації правил продукцій; виконує дію з обраної процедури [11].

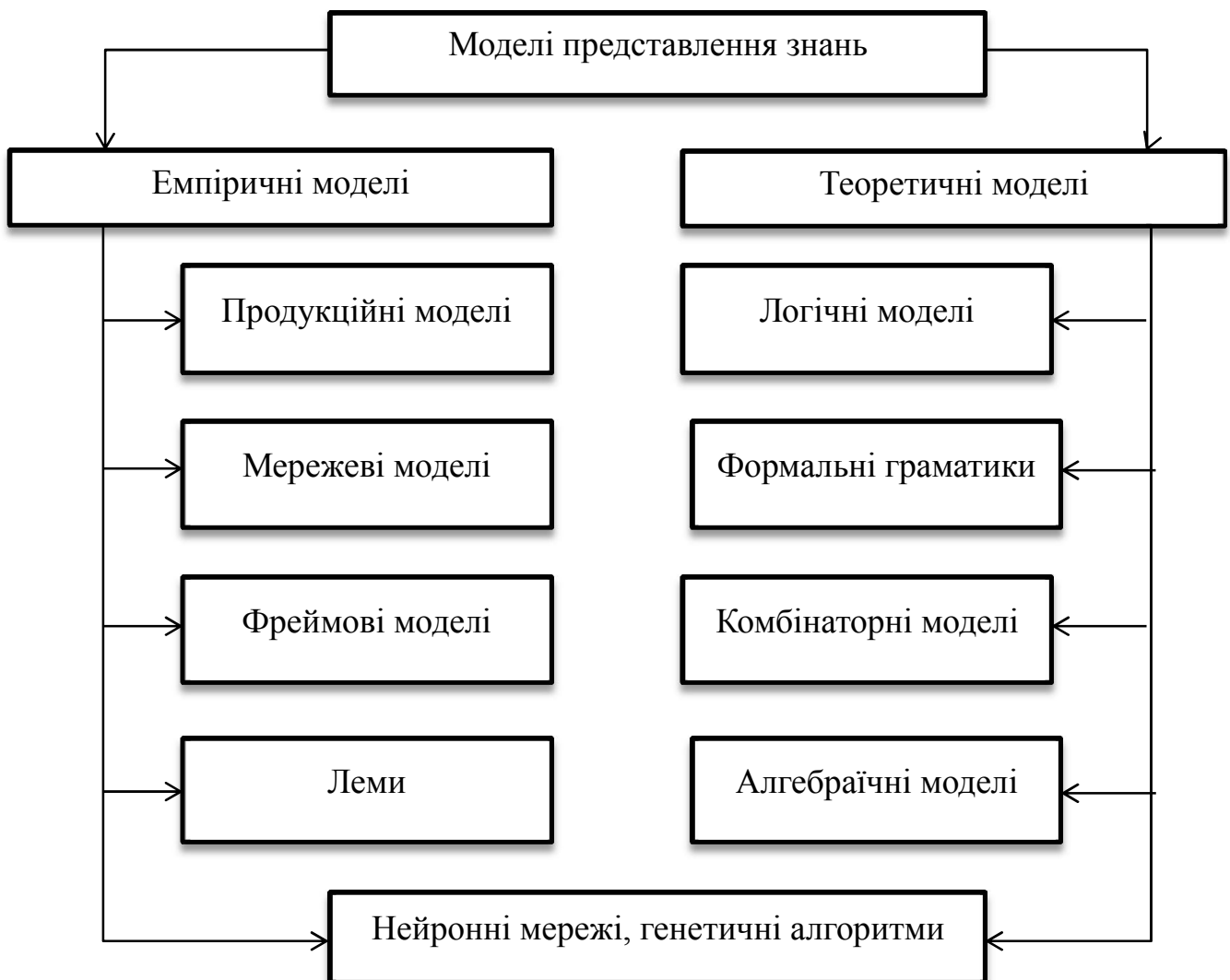


Рисунок 1.4 – Класифікація моделей представлення знань

Системи обробки знань, що використовують продукційну модель отримали назву «продукційних систем». До складу експертних систем продукційного типу входять база правил (знань), робоча пам'ять і інтерпретатор правил, який реалізує певний механізм логічного висновку. Будь-яке продукційне правило, яке міститься в базі знань, складається з двох частин: антецедент і консеквента. Антецедент являє собою посилку правила (умовну частину) і складається з елементарних пропозицій, з'єднаних логічними зв'язками «і», «або». Консеквент (висновок) включає одне або кілька пропозицій, які висловлюють або деякий факт, або вказівка на певну дію, підлягає виконанню. Продукційні правила прийнято записувати у вигляді антецедент-консеквент [12].

Запис продукційних правил відображається наступним чином:

IF розмір_клітини = збільшений AND форма_клітини = циліндрична

OR розмір_клітини = збільшений AND форма_клітини = кубічна

OR розмір_клітини = гігантський AND форма_клітини = кубічна

OR розмір_клітини = гігантський AND форма_клітини = циліндрична

OR розмір_клітини = гігантський AND форма_клітини = округла

THEN діагноз = рак.

Будь-яке правило складається з одного або декількох пар «атрибут-значення». В робочій пам'яті систем, заснованих на продукційних моделях зберігаються пари атрибут-значення, істинність яких встановлена в процесі вирішення конкретної задачі для деякого моменту часу. Вміст робочої пам'яті змінюється в процесі вирішення задачі. Це відбувається в міру спрацьовування правил. Правило спрацьовує, якщо при зіставленні фактів, що містяться в робочій пам'яті, з антецедентом аналізованого правила має місце збіг, при цьому укладення спрацьовували правила заноситься в робочу пам'ять. Тому в процесі логічного висновку обсяг фактів в робочій пам'яті, як правило, збільшується (зменшуватися він може в тому випадку, якщо дія якого-небудь правила полягає у видаленні фактів з робочої пам'яті). У

процесі логічного висновку кожне правило з бази правил може спрацювати тільки один раз.

Існують два типи продукційних систем – з «прямими» і «зворотними» висновками. Прямі висновки реалізують стратегію «від фактів до висновків». При зворотних висновках висуваються гіпотези імовірнісних висновків, які можуть бути підтвержені або спростовані на підставі фактів, що надходять в робочу пам'ять [12]. Існують також системи з двонаправленими висновками, а саме:

1) мережеві моделі (або семантичні мережі) представляються у вигляді графу, що відображає сенс цілісного образу. Вузли графа відповідають поняттям і об'єктам, а дуги – відповідністю між об'єктами. Володіє тим недоліком, що однозначного визначення семантичної мережі в даний час не існує;

2) фреймова модель ґрунтується на такому понятті як фрейм (англ. frame – рамка, каркас). Фрейм – це структура даних для представлення деякого концептуального об'єкта. Інформація, що відноситься до фрейму, міститься в складових його слотах. Слоти можуть бути термінальними або бути самими фреймами, утворюючи цілу ієрархічну мережу.

3) леми – доведені твердження, корисні не самі по собі, а для доказу інших тверджень.

До групи емпіричного підходу можна включити нейронні мережі й генетичні алгоритми, що відносяться до біонічного (ґрунтуються на припущенні про те, що якщо в штучній системі відтворити структури і процеси людського мозку, то й результати вирішення завдань такою системою будуть подібні результатами, одержуваною людиною) напрямку штучного інтелекту. Особливістю моделей цього типу є широке використання евристики, що в кожному випадку вимагає доказу правильності одержуваних рішень [13].

Другий підхід можна визначити як теоретично обґрунтований, що

гарантує правильність рішень. Він в основному представлений моделями, заснованими на формальній логіці (числення висловів, числення предикатів), формальних граматиках, комбінаторними моделями, зокрема моделями кінцевих проєктивних геометрій, теорії графів, тензорними і алгебраїчними моделями. У рамках цього підходу до теперішнього часу вдавалося вирішувати тільки порівняно прості завдання з вузької предметної області.

Логічні моделі включають класичний механізм подання знань в системах, де використовуються предикати. У системах, заснованих на численні предикатів, знання представляються за допомогою переказу тверджень про об'єкти деякої предметної області у формули логіки предикатів і додавання їх як аксіом в систему. Знання відображаються сукупністю таких формул, а отримання нових знань зводиться до реалізації процедур логічного висновку. Проте дійсність не вкладається в рамки класичної логіки, бо людська логіка, застосовувана при роботі з неструктурованими знаннями – це інтелектуальна модель з нечіткою структурою. При використанні нечіткої логіки часто застосовуються два методи логічного висновку: прямий і зворотний метод. Переваги даного методу полягають в класичному апараті математичної логіки, методи якої добре вивчені та обґрунтовані, маються досить ефективні процедури виведення. База знань призначена для зберігання великої кількості аксіом, з яких за правилами виведення можна отримувати інші знання [14]. Основний недолік: логіки, яка адекватно відображає людське мислення, ще не створено.

При розгляді класу нейронних мереж необхідно звернути увагу на новий науково-практичний напрямок – створення нейрокомп'ютера, ЕОМ нового покоління, який здатний до самоорганізації, навчанню і імітує деякі здібності людського мозку з обробки інформації. Результатами з'явилися подання знань, засновані на масовій паралельній обробці, швидкому пошуку великих обсягів інформації і здатності розпізнавати зразки. Технології, спрямовані на досягнення цих результатів відносяться до нейронних

обчислень або штучних нейронних мереж.

Нейронні мережі дозволяють вибрати відповідні параметри для генетичних алгоритмів (параметри схрещення мутації); фундамент нейронних мереж будується на нечітких системах, які в результаті набувають здібність до навчання [15].

Генетичні алгоритми застосовуються при роботі програмного забезпечення в системах штучного інтелекту, оптимізації штучних нейронних мереж та інших галузях. З їх допомогою вирішуються задачі для яких раніше використовувались тільки нейронні мережі. В даному випадку, генетичні алгоритми виступають у ролі незалежного від нейронних мереж альтернативного методу, призначені для вирішення тієї ж задачі.

Генетичний алгоритм представляє собою метод, який відбиває природню еволюцію методів вирішення проблем, а в першу чергу – задачі оптимізації. Генетичні алгоритми – процедури пошуку, засновані на механізмах природнього відбору і наслідування [16]. В них використовується еволюційний принцип виживання найбільш пристосованих осіб та відрізняються від традиційних методів деякими елементами:

- 1) обробляються не значення параметрів самої задачі, а їх закодована форма;
- 2) здійснюється пошук рішень виходячи не з єдиної точки, а деякої множини;
- 3) використовується цільова функція, а не її похідна;
- 4) застосовується ймовірність, а не детерміновані правила вибору.

1.3 Основні поняття нечіткої логіки

В спробі класифікації людських знань дослідники зіткнулись з проблемою, яка викликала труднощі використання традиційного математичного апарату для їх опису. Існує цілий клас опису об'єкту, який оперує кількісними

характеристиками (багато, мало, сильний та інші).

Значення терміну нечіткість багатозначне. Важко претендувати на вичерпне визначення цього поняття, тому розглянемо лише основні його компоненти, до яких відносяться наступні:

- 1) недетермінованість висновків;
- 2) багатозначність;
- 3) ненадійність;
- 4) неповнота;
- 5) неточність.

Недетермінованість висновків. Це характерна риса більшості систем штучного інтелекту. Недетермінованість означає, що заздалегідь шлях вирішення конкретного завдання в просторі її станів визначити неможливо. Тому в більшості випадків методом проб і помилок вибирається деяка ланцюжок логічних висновків, що узгоджуються з наявними знаннями, а в разі якщо вона не приводить до успіху, організується перебір з поверненням для пошуку іншої ланки і т.д. Такий підхід передбачає визначення деякого початкового шляху. Для вирішення подібних завдань запропоновано багато евристичних алгоритмів.

Багатозначність інтерпретації – звичайне явище в задачах розпізнавання. При розуміння природної мови серйозними проблемами стають багатозначність змісту слів, їх підпорядкованості, порядку слів у реченні і т. П. Проблеми розуміння значення виникають в будь-якій системі, що взаємодіє з користувачем на природній мові. Розпізнавання графічних образів також пов'язане з вирішенням проблеми багатозначною інтерпретації. При комп'ютерній обробці знань багатозначність необхідно усувати шляхом вибору правильної інтерпретації, для чого розроблені спеціальні методи, наприклад, метод релаксації, призначений для систематичного усунення багатозначності при інтерпретації зображень.

Ненадійність знань і висновків. Ненадійність знань означає, що для

оцінки їх достовірності не можна застосувати двобальну шкалу (1 – абсолютно достовірні; 0 – недостовірні знання). Для більш тонкої оцінки достовірності знань застосовується імовірнісний підхід, заснований на теоремі Баєса, і інші методи. Наприклад, в експертній системі MYCIN, призначеної для діагностики та вибору методу лікування інфекційних захворювань, розроблений метод виведення з використанням коефіцієнтів упевненості. Широке застосування на практиці отримали нечіткі висновки, що будуються на базі нечіткої логіки, що веде своє походження від теорії нечітких множин.

Неповнота знань і немонотонна логіка. Абсолютно повних знань не буває, оскільки процес пізнання нескінченний. У зв'язку з цим стан бази знань повинно змінюватися з плином часу. На відміну від простого додавання інформації, як в базах даних, при додаванні нових знань виникає небезпека отримання суперечливих висновків, тобто висновки, отримані з використанням нових знань, можуть спростовувати ті, що були отримані раніше. Ще гірше, якщо нові знання будуть перебувати в суперечності зі «старими», тоді механізм виведення може стати непрацездатним. Багато експертних системи першого покоління були засновані на моделі закритого світу, обумовленої застосуванням апарату формальної логіки для обробки знань. Модель закритого світу передбачає жорсткий відбір знань, що включаються в базу, а саме: БЗ заповнюється виключно вірними поняттями, а все, що ненадійно або невизначено, свідомо вважається хибним. Іншими словами, все, що відомо базі знань, є істиною. Така модель має обмежені можливості подання знань і таїть в собі небезпеку отримання протиріч при додаванні нової інформації. Тим не менш, ця модель досить поширена; наприклад, на ній базується мову PROLOG. Недоліки моделі закритого світу пов'язані з тим, що формальна логіка виходить з передумови, згідно з якою набір визначених у системі аксіом (знань) є повним (теорія є повною, якщо кожен її факт можна довести, виходячи з аксіом цієї теорії). Для повного набору знань справедливості раніше отриманих висновків не порушується з додаванням нових фактів. Ця властивість логічних висновків

називається монотонністю. На жаль, реальні знання, які закладаються в експертні системи, дуже рідко бувають повними.

Неточність знань. Відомо, що кількісні дані (знання) можуть бути неточними, при цьому існують кількісні оцінки такої неточності (довірчий інтервал, рівень значущості, ступінь адекватності). Розвиток досліджень в області нечіткої математики привело до появи нечіткої логіки та нечітких висновків, які виконуються з використанням знань, представлених нечіткими множинами, нечіткими відносинами, нечіткими відповідниками [17].

Дані характеристики зазвичай розмиті і не можуть бути однозначно інтерпретовані, однак містять необхідну інформацію. Крім того, в задачах, які вирішуються інтелектуальними системами, часто доводиться використовувати знання, які не можуть бути представлені, як повністю істинні чи фальшиві (логічні true чи false або 0 чи 1). Існують знання, достовірність яких виражається проміжним числом, наприклад 0,2.

З точки зору обробки нечітких множин навіть в теорії ймовірності, яку часто протиставляють теорії нечітких систем не можна отримати необхідні результати, доки експерт не дослідить в повній мірі вихідні дані, оцінить загальну структуру проблеми і не виключить сумнівні дані або не прийме інші суб'єктивні рішення. Тому, на гіпотези та посилання не можна посилалися як не на суб'єктивну думку.

Питання обробки нечітких продукційних правил знаходиться на одному рівні з питанням як ввести в науку і техніку суб'єктивізм людини і тут не обійтися без нечітких множин. Цей математичний метод створено для того, щоб представляти змістовні нечіткості слів людини – це унікальний метод з точки зору представлення можливості математичної обробки суб'єктивних даних [17].

Можливість математично описати зміст та суть слів означає, що появились нові засоби, які дозволяють задовільнити три необхідні умови реалізації, а саме: завдяки їм, можна очікувати, що штучний інтелект зможе розуміти нечіткість і зміст виражених потреб та його спілкування з людиною

стане більш природнім. Крім чого, можна буде представити у вигляді нечітких тез навіть ті нечіткі макрознання, як здоровий смисл та звузити спеціальну область. Такі умови повисять практичну цінність штучного інтелекту. На останок, деякі дані, які отримуються від людини можуть пригодитися, як якісні логічні відношення між макрознаннями, тому в будь-якому степені логічні виводи присутні в людському міркуванні [18].

Для вирішення такого класу завдань, на початку 70-тих років американський математик Лотфі Заде запропонував формальний апарат нечіткої алгебри (fuzzy) і нечіткої логіки та ввів одну з галузей головних понять в нечіткій логіці – поняття лінгвістичної змінної [19].

Лінгвістична змінна – це змінна, значення якої визначається набором вербальних (словесних) характеристик деякої властивості. Лінгвістична змінна це багато нечітких змінних, вона використовується для того щоб дати словесний опис деякому нечіткому числу, отриманому в результаті деякими операціями.

Значення лінгвістичної змінної визначається через так звані нечіткі множини, які в свою чергу визначені на деякому базовому наборі чи базовій шкалі, яка має розмірність [20].

Звичайну підмножину можна було б ототожнити з його характеристичної функцією. Цього математично неможливо зробити, оскільки для завдання функції необхідно спочатку поставити багато. Нечітка ж підмножина з формальної точки зору можна ототожнити з його функцією належності. Однак термін «нечітка підмножина» переважає при побудові математичних моделей реальних явищ.

Методи нечітких множин дозволяють вибирати параметри генетичних алгоритмів, а також коефіцієнти, які визначають швидкість навчання нейронних мереж.

Нечіткі множини визначаються через функцію належності, тобто нечіткі продукційні правила не містять будь-яких не роз'яснень. У випадку нечіткої логіки, можна отримати необмежену кількість операцій, тому не виникає

потреби записати всі операції за допомогою деякого числа базових операцій, враховуючи це, необхідно обмежитися найбільш важливими операціями [21].

Нечіткі множини визначаються через деяку базову шкалу V і функцію належності - $\mu(x)$, $x \in V$, яка приймає значення на інтервалі $[0 \dots 1]$. Таким чином, нечітка множина V – сукупність пар типу $(x, \mu(x))$, де $x \in V$. Часто зустрічається такий запис:

$$V = \sum_{i=1}^n \frac{x_i}{\mu(x_i)}, \quad (1.1)$$

де (x_i) – i -те значення базової шкали.

Функція належності визначає суб'єктивний степінь впевненості експерта в тому, що дане конкретне значення базової шкали відповідає значенню нечіткої множини [22].

Для операції з нечіткими знаннями, які виражаються за допомогою лінгвістичних змінних, існує багато способів. Дані способи являються евристичними, для прикладу функція «АБО» часто задається як:

$$\mu(x) = \max(\mu_1(x), \mu_2(x)), \text{ або } \mu(x) = \mu_1(x) + \mu_2(x) - \mu_1(x) * \mu_2(x)$$

Посилення чи послаблення степеню в лінгвістичному понятті досягається введенням спеціальних квантифікаторів.

Для виводу на нечітких множинах використовуються спеціальні відношення та операції над ними. Одним з перших застосувань теорії нечітких множин стало використання коефіцієнтів впевненості для виводу рекомендації медичної системи MYCIN. Даний метод використовує декілька евристичні прийоми, став прикладом обробки нечітких знань, які з'явилися в наступних системах. На даний момент, більшість інструментальних засобів розробки систем, заснованих на знаннях, включають елементи роботи з нечіткими множинами, крім того, розроблені спеціальні програмні середовища реалізації так званого нечіткого виводу, на приклад Fuzzy Rules [22].

Для побудови бази нечітких правил необхідно навчальну вибірку даних в вигляду множини пар чисел:

$$(x_1(i), x_2(i), d(i)), \quad i = 1, 2, \quad (1.2)$$

де $x_1(i)$, $x_2(i)$ – сигнали, які подаються на вхід модуля нечіткого керування, а $d(i)$ – очікуване значення вихідного сигналу.

Задача продукційних правил полягає в формуванні таких нечітких правил, щоб побудованих на їх основі модуль керування при отриманні вхідних сигналів генерував коректні вихідні сигнали. Тому, нам необхідно поділити простір вхідних і вихідних сигналів на області та створити відповідні їм функції належності. Побудувати нечіткі правила на основі навчальної вибірки, створити таблицю для запису бази правил та таблицю степенів істинності правил. Приписати кожному правилу степінь істинності та сформулювати відповідні правила, створити базу нечітких правил [23].

1.4 Аналіз програмних засобів представлення знань

Система Matlab, розроблена і постійно оновлюється компанією Math Works Inc. (США), є однією з найбільш відомих систем комп'ютерної математики.

Використання системи Matlab і зв'язаних з нею методик моделювання і процедур виконання числових розрахунків стало стандартом для широкого круга спеціалістів різноманітних областей науки, техніки, освіти та інших. Вміст спеціальних засобів нечіткого моделювання, система Matlab дозволяє вивчати весь комплекс дослідження по розробці і застосуванню нечітких моделей. Саме через ці причини система Matlab була вибрана у якості програмного засобу, в рамках якого можна реалізувати концепції нечітких множин і процедур нечітких продукційних правил [24].

Система Matlab – математична лабораторія, представляє собою інтегроване програмне середовище для виконання числових обчислень, комп'ютерного моделювання і обчислювальних експериментів, які в тій чи іншій мірі охоплюють області класичної та сучасної математики, спектр

інженерних додатків.

Архітектурно система Matlab складається з базової програми та декількох десятків пакетів розширення, які в своїй сукупності забезпечує виключно широкий діапазон задач, які вирішуються. Інтеграція всіх засобів в єдиному робочому середовищі забезпечують необхідну гнучкість використання сотень вбудованих функцій, які реалізують математичні процедури та обчислювальні алгоритми [25].

Нечітке моделювання в середовищі Matlab здійснюється за допомогою використання пакету розширень Fuzzy Logic Toolbox, в якому реалізовані десятки функцій нечіткої логіки і нечіткого виводу.

Matlab є мовою високого рівня та інтерактивне середовище для чисельних обчислень, візуалізації та програмування. З використання Matlab можна аналізувати дані, розробляти алгоритми і створювати моделі і додатки. Мова, інструменти вбудованих математичних функцій дозволяють досліджувати кілька підходів і досягти рішення швидше, ніж з електронними таблицями або традиційними мовами програмування, таких як C / C ++ або Java. Matlab застосовується для широкого спектру додатків, включаючи обробку сигналів і повідомлень, зображень і обробки відео, систем управління, тестування і вимірювань і обчислювальної біології.

Після запуску системи Matlab на екрані з'являється основне діалогове вікно для взаємодії з системою, що містить декілька діалогових підвікон. Одне з них має заголовок «Command window» - це командне діалогове вікно, в якому користувач набирає команди, а Matlab видає результати виконання команд, що містять графічний висновок, видаються в одне або кілька графічних вікон.

Значення всіх створюваних в командному вікні змінних зберігаються у виділеній області пам'яті, яка називається робочим простором командного вікна або основним робочим простором. Крім основного робочого простору Matlab створює і в потрібний час видаляє робочі простору викликаються функцій. У цих робочих просторах розміщуються внутрішні змінні функцій. Імена всіх

змінних, що знаходяться в даний момент в основному робочому середовищі відображаються в підвікні «Workspace» [26].

Matlab підтримує роботу з різними типами даних, а саме тип «double», що представляє матриці, елементи яких – дійсні або комплексні числа з плаваючою комою подвійної точності, інший тип- «char» - представляє масиви символів.

У будь-якій мові програмування, у тому числі в мові, надає система Matlab, є спеціальні конструкції, призначені для керування порядком виконання команд. Такі конструкції іноді називають керуючими операторами.

У Matlab до них відносяться:

- 1) умовний оператор «if»;
- 2) оператор циклу «while»;
- 3) оператор циклу з параметром «for»;
- 4) оператор вибору «switch» [27].

Програмою на мові Matlab ми називатимемо послідовність команд, записану у файлі. Щоб система приймала програму, у файлу має бути розширення .m, тому програми в Matlab часто називають m-файлами.

До програм можна звертатися з командного вікна і з інших програм. При першому зверненні до програмі Matlab шукає її на диску по імені файлу (без розширення). В першу чергу, пошук проводиться в поточній папці.

M –файл з програмою-функцією може містити опис не однієї, а кількох функцій. Ім'я першої функції повинне збігатися з ім'ям файлу. Тільки ця функція (основна) доступна ззовні. Кожна з них може бути викликана або з основної функції, або з іншої підфункції того ж самого m-файла. Кінець опису основної функції або підфункції ніяким спеціальним ніяк не позначається: опис підфункції закінчується або в кінці файлу, або безпосередньо перед початком опису наступній підфункції перед рядком, що містить ключове слово «function»).

Існує інший спосіб написання m-файлів, що містять кілька функцій.

Функції можна вкладати всередину інших функцій. У цьому випадку

кожна з функцій, описаних у файлі повинна закінчуватися ключовим словом «end» [28].

Програма-сценарій («script») – найпростіший тип програми. Програма-сценарій може використовувати не тільки створювані їй самі змінні, але і використовувати всі змінні в робочому просторі командного вікна. Створені змінні так само розташовуються в цьому робочому просторі. Отримати до них доступ можна за допомогою команди «keyboard», яку потрібно записати в програму [29].

Fuzzy Logic Toolbox надає функції Matlab, додатки і блок Simulink для аналізу, проектування, моделювання та систем, заснованих на нечіткій логіці. Продукт дає змогу використання проектування систем нечіткого виводу в повному обсязі. Функції призначені для багатьох поширених методів, в тому числі нечіткої кластеризації та адаптивного навчання з нечіткою логікою.

Панель інструментів дозволяє моделювати складні системи поведінки за допомогою простих логічних правил, а потім реалізувати ці правила в нечіткій системі виводу. Крім того, ви можливе використання нечіткого виводу блоків в Simulink і моделювання нечітких систем в рамках комплексної моделі всієї динамічної системи [30].

Ключові особливості:

- 1) графічний інтерфейс для інтерактивного покрокового проектування нечітких систем;
- 2) функції для створення експертних систем на основі нечіткої логіки;
- 3) підтримка логіки І, АБО і НЕ в налаштуваннях правил;
- 4) стандартні типи експертних систем нечіткої логіки (Mamdani, Sugeno);
- 5) функції для нейроадаптивної нечіткої кластеризації з навчанням;
- 6) включення нечітких систем в Simulink-моделі;
- 7) генерація С-коду і незалежних додатків, що реалізують системи нечіткої логіки.

При нечіткому моделюванні часто використовують базу знань у форматі

Mamdani, в якій антецедента і консеквентом правил задані нечіткими множинами, наприклад «низький»(low), «середній»(average), «високий»(high) і т.п. Нечіткі правила в такому форматі запропоновані, на основі якої Е. Мамдані і С. Ассіліан розробили перший нечіткий контролер.

Нечіткі моделі Mamdani прозорі, їх структура змістовно інтерпретується в термінах, зрозумілих не тільки розробникам з високою математичною кваліфікацією, а і замовникам – лікарям, економістам, менеджерам. Прозорість нечітких моделей Mamdani – одна з головних переваг, завдяки якій нечіткі технології успішно конкурують з іншими методами, особливо для тих прикладних задач, де можливість змістовної інтерпретації важливіше точності моделювання [31].

Для підвищення точності нечіткої моделі навчають, тобто ітераційно міняють її параметри для мінімізації відхилення результатів логічного висновку від експериментальних даних. Налаштовують як ваги правил, так і функції належності нечітких термів. Навчання нечіткої моделі Mamdani являє собою завдання нелінійної оптимізації, дослідженню якої присвячено великій кількості теоретичних і прикладних робіт, у яких основний акцент робиться на досягнення максимальної точності навчання нечіткої моделі. При цьому настроюються параметри іноді змінюються настільки сильно, що виникають складності змістовної інтерпретації нечіткої моделі. Таким чином, задача покращення приводить до втрати важливої конкурентної переваги – прозорості нечіткої моделі. Якщо прозорість моделі друго-ступенева, то при ідентифікації залежностей розумніше використовувати інші (чіткі) методи, адекватність яких зазвичай краще [32].

Модуль fuzzy дозволяє будувати нечіткі системи двох типів – Mamdani і Sugeno. У системах типу Mamdani база знань, яка складається з правил виду: «якщо $x_1 = \text{низький}$ і $x_2 = \text{середній}$, то $y = \text{високий}$ ». У системах типу Sugeno база знань складається з правил виду: «якщо $x_1 = \text{низький}$ і $x_2 = \text{середній}$, то $y = a_0 + a_1x_1 + a_2x_2$ ». Таким чином, основна відмінність

між системами Mamdani і Sugeno полягає в різних способах завдання значень вихідної змінної в правилах, що утворюють базу знань. У системах типу Mamdani значення вихідної змінної задаються нечіткими термами, в системах типу Sugeno – як лінійна комбінація вхідних змінних [33].

Simulink – це графічне середовище імітаційного моделювання, що дозволяє за допомогою блок-діаграм у вигляді направлених графів, будувати динамічні моделі, включаючи дискретні, безперервні і гібридні, нелінійні і розривні системи.

Інтерактивне середовище Simulink, дозволяє використовувати вже готові бібліотеки блоків для моделювання електросилових, механічних та гідравлічних систем, а також застосовувати розвинений модельно-орієнтований підхід при розробці систем управління, засобів цифрового зв'язку і пристроїв реального часу.

Додаткові пакети розширення Simulink дозволяють вирішувати весь спектр завдань від розробки концепції моделі до тестування, перевірки, генерації коду і апаратної реалізації. Simulink інтегрований в середовище Matlab, що дозволяють використовувати вбудовані математичні алгоритми, потужні засоби обробки даних.

Simulink Library Browser (засіб перегляду бібліотеки Simulink) містить в собі бібліотеку блоків найбільш часто використовуваних для моделювання систем.

У цю бібліотеку входять:

- 1) блоки безперервної і дискретної динаміки, такі як «Integrator» (інтегратор) і «Unit delay» (ланка затримки);
- 2) алгоритмічні блоки, такі як «Sum» (суматор), «Product» (категорія), «Lookup table» («довідкова таблиця»);
- 3) структурні блоки, такі як «Mux» (мультиплексор), «Switch» (перемикач), «Bus selector» (селектор шини) [35].

Програмне забезпечення Matlab найбільш широко використовується як

чисельний обчислювальний пакет, проте є аналоги, які мають багато таких самих функцій, як Matlab. Одним із представників є GNU Octave – найбільш сумісний з Matlab через його можливості та подібність синтаксису.

GNU Octave - вільна система для математичних обчислень [00], що використовує сумісну з Matlab мову високого рівня. Octave представляє інтерактивний командний інтерфейс для вирішення лінійних і нелінійних математичних задач, а також проведення інших численних експериментів. Крім того, Octave можна використовувати для пакетної обробки. Мова Octave оперує арифметикою дійсних і комплексних скалярів і матриць, має розширення для вирішення лінійних алгебраїчних задач, знаходження коренів систем нелінійних алгебраїчних рівнянь, роботи з поліномами, рішення різних диференціальних рівнянь, інтегрування систем диференціальних та диференціально-алгебраїчних рівнянь першого порядку, інтегрування функцій на кінцевих і нескінченних інтервалах.

Octave написаний з врахуванням сумісності з Matlab і реалізує багато його можливостей:

- 1) матриці в якості основних типів даних;
- 2) вбудована підтримка комплексних чисел;
- 3) потужні вбудовані математичні функції і великі бібліотеки функцій;
- 4) розширюваність, завдяки можливості створення призначених для користувача функцій.

Таким чином, для використання в наукових дослідженнях Octave має багато функцій, які роблять його альтернативою Matlab.

1.5 Постановка завдання дипломної роботи

Основним завданням дипломної роботи є розробка алгоритмів побудови нечітких продукційних правил на основі аналізу біомедичних зображень.

Для досягнення поставленої мети виконуються наступні завдання:

- 1) аналіз бази даних цитологічних та гістологічних зображень;
- 2) дослідити відомі моделі представлення знань;
- 3) проаналізувати нечіткі моделі представлення знань;
- 4) провести аналіз програмних засобів представлення знань;
- 5) визначити функції належності на основі кількісних ознак мікрооб'єктів, отриманих від експерта;
- 6) побудувати функції належності в середовищі Fuzzy Logic Toolbox;
- 7) побудувати нечіткі продукційні правила в середовищі Fuzzy Logic Toolbox;
- 8) здійснити експериментальні дослідження передракових та ракових станів молочної залози.

2 АЛГОРИТМИ ПОБУДОВИ НЕЧІТКИХ ПРОДУКЦІЙНИХ ПРАВИЛ

2.1 Нечітка модель представлення знань та її модифікації

Продукційна модель представлення знань має такий вигляд:

Якщо умова (антецедент), то висновок (консеквент).

Вище наведена форма запису виражається як та, що стосується дедуктивної форми запису правил. Правило такого запису виражає логічний висновок, тому якщо ми знаємо факт (припущення, гіпотеза, що передує), то ми можемо зробити висновок, чи виводимо ще один факт – висновок. Ця форма представлення знань, характеризується як представлення неглибокого знання, цілком доречного в контексті лінгвістики, оскільки він висловлює людські емпіричні і евристичні знання в рідній мові спілкування. Проте він не дозволяє оволодіти глибшими формами знань, які зазвичай асоціюються з інтуїцією, функцією та поведінкою об'єктів навколо нас. Тому, ці форми знання не завжди зводиться до лінгвістичних фраз або уявлень, такі форми представлення знань глибші, називаються індуктивними [36].

Нечітка система на основі правил є найбільш корисною при моделюванні деяких складних систем, які можна спостерігати у суспільстві, тому що воно використовує лінгвістичні змінні, як їх передумови і наслідки. Лінгвістичні змінні можуть бути природним чином представлено в нечітких множинах і логічними зв'язками цих множин. Використовуючи основні властивості і операції, що визначається для нечітких множин, будь-яка структура сполуки правила може бути розкладено і зведено до числа простих канонічним правил.

Ці правила засновані на природних уявлень і мовних моделях, які засновані на основі нечітких множин і нечіткої логіки. Нечіткий рівень розуміння і опису складної системи виражається у вигляді набору обмежень на виході на основі певних умов вводу. Обмеження, як правило, моделюються нечіткими множинами та зв'язками. Такі обмеження пов'язані з лінгвістичними зв'язками, наприклад, «І», «АБО», «АБО-НЕ» [37].

Приклад канонічної форми запису:

Rule 1: IF condition C1, THEN restriction R1

Rule 2: IF condition C2, THEN restriction R2

...

Rule r: IF condition Cr, THEN restriction Rr

В загальному, три основні форми існують за рахунок лінгвістичних змінних. Ці форми є: оператор присвоєння, умовний оператор та безумовний оператор [38]. Їхні приклади наведено нижче.

Оператор присвоєння

$x =$ розмір великий,

Форма = овальна,

$x \approx s$,

x не є великим і не є дуже малим,

Форма = кубічна,

Розмір = малий,

Діагноз – нормальна тканина.

2) Умовний оператор

IF розмір середній THEN діагноз нормальна тканина,

IF x розмір великий THEN ракова клітина,

IF x розмір великий THEN y розмір середній ELSE y не є малим розміром.

3) Безумовний оператор

Форма овальна,

Розмір не великий,

Розмір не малий,

Діагноз – нормальна тканина

Оператор присвоєння обмежує значення змінної до певної кількості. Безумовний оператор може розглядатися як умовне обмеження з їхніми IF умовами будучи універсумом дискусу усіх існуючих вхідних умов, які завжди вірні. Безумовний оператор типу «вихід є низьким» може еквівалентно бути

записаним як:

```
IF будь-які умови THEN вивід низький, чи  
IF будь-що THEN низький.
```

Отже, засновані на нечітких правилах системи можуть бути описані використовуючи умовний оператор. Цей оператор може бути змодельований як нечіткі умовні оператори як:

```
IF умова  $C^1$  THEN обмеження  $R^1$ 
```

Безумовний оператор може бути записаний в формі:

```
 $R^1$  : вивід  $B^1$ 
```

```
AND
```

```
 $R^2$  : вивід  $B^2$ 
```

```
AND
```

де B^1 та B^2 – нечіткі консеквенти.

Лінгвістичний виклад, який виражається людським міркуванням залучає з'єднувати структури правил. Як приклад, розглядається система заснована на правилах для виявлення ракових та передракових станів молочної залози [39], які можуть бути сформовані наступним чином:

```
IF розмір середній
```

```
THEN діагноз нормальна тканина.
```

```
IF форма клітини кубічна
```

```
THEN
```

```
IF розмір клітини малий
```

```
THEN діагноз проліферативна мастопатія
```

```
ELSE
```

```
IF (розмір малий) AND (форма циліндрична)
```

```
THEN діагноз проліферативна мастопатія.
```

Використовуючи прості властивості та операції, які визначають нечіткі множини, будь-яке правило структури з'єднання може бути розкладене на декілька простих правил канонічної форми.

Приклад канонічної форми запису:

Rule 1: IF condition C1, THEN restriction R1

Rule 2: IF condition C2, THEN restriction R2

...

Rule r: IF condition Cr, THEN restriction Rr

Ці правила засновані на мові уявлення та моделі, які власне самі засновані на нечітких множинах та нечіткій логіці.

Множина кон'юнктивних антецедентів

$$\text{IF } x \text{ is } A^1 \text{ and } A^2 \dots \text{ and } A^L \text{ THEN } y \text{ is } B^S$$

Припускаючи, що нова нечітка підмножина A^S , то:

$$A^S = A^1 \cap A^2 \cap \dots \cap A^L$$

виражається за допомогою функції належності:

$$\mu_{A^S}(x) = \min[\mu_{A^1}(x), \mu_{A^2}(x), \dots, \mu_{A^L}(x)]$$

заснований на основі визначення стандартної нечіткої операції перетину, правило з'єднання можна переписати у вигляді:

$$\text{IF } A^S \text{ THEN } B^S$$

Множина диз'юнктивних антецедентів

$$\text{IF } x \text{ is } A^1 \text{ OR } x \text{ is } A^2 \dots \text{ OR } x \text{ is } A^L \text{ THEN } y \text{ is } B^S$$

повинна бути записано як:

$$\text{IF } x \text{ is } A^S \text{ THEN } y \text{ is } B^S,$$

де нечітка множина A^S визначається як:

$$A^S = A^1 \cup A^2 \cup \dots \cup A^L$$

$$\mu_{A^S}(x) = \max[\mu_{A^1}(x), \mu_{A^2}(x), \dots, \mu_{A^L}(x)],$$

який заснований на визначенню стандартної нечіткої операції об'єднання.

Умовний оператор АБО-НЕ та ЯКЩО-НЕ

$$\text{IF } A^1 \text{ THEN } (B^1 \text{ ELSE } B^2)$$

може бути розкладений на дві прості канонічні форми правил, які поєднуються АБО.

$$\text{IF } A^1 \text{ THEN } B^1$$

OR
 IF NOT A^1 THEN B^2
 IF A^1 (THEN B^1) UNLESS A^2

може бути розкладено як:

IF A^1 THEN B^1
 OR
 IF A^2 THEN NOT B^1
 IF A^1 THEN (B^1 ELSE IF A^2 THEN (B^2))

може бути записано в наступній формі:

IF A^1 THEN B^1
 OR
 IF NOT A^1 AND A^2 THEN B^2

Вкладені ЯКЩО-ТО правила

IF A^1 THEN (IF A^2 THEN B^1)

може бути записано у формі

IF A^1 AND A^2 THEN B^1

Коли правила розкладені на серію канонічних форм, кожна з форм є імплікацією і ми можемо зменшити правила і серію зв'язків [40].

Більшість систем, засновані на правилах, які включають більше одного правила. Процес отримання загальної консеквенти (висновку) від окремих наслідків внесених кожним правилом в правилі бази, називається агрегацією правил. При визначенні стратегії агрегації існує два простих крайніх випадки:

1. Кон'юнктивна система правил. У разі системи правил, які повинні спільно задовольнятися, правила пов'язані «І» зв'язками. В цьому випадку агрегований вихід (як наслідок), у виявляється за допомогою нечіткого перетину всіх індивідуальних правил наслідок, y^i , де *where* $i = 1, 2, \dots, r$, як

$$y = y^1 \text{ and } y^2 \text{ and } \dots \text{ and } y^r$$

або

$$y = y^1 \cap y^2 \cap \dots \cap y^r,$$

який визначено функцією належності

$$\mu_y(y) = \min(\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^r}(y)) \text{ для } y \in Y \text{ [41].}$$

2) Диз'юнктивна система правил. Для випадку у диз'юнктивній системі правил, де потрібно задовільнити щонайменше одне правило, правила пов'язані за допомогою зв'язків «АБО». В цьому випадку сумарний вихід знайдений нечітким об'єднанням всіх окремих частин правила, в якості:

$$y = y^1 \text{ or } y^2 \text{ or } \dots \text{ or } y^r$$

або

$$y = y^1 \cup y^2 \cup \dots \cup y^r ,$$

який визначено функцією належності

$$\mu_y(y) = \max(\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^r}(y)) \text{ для } y \in Y \text{ [42].}$$

2.2 Побудова функцій належності лінгвістичних змінних

Нечіткою множиною (fuzzy set) \tilde{A} на універсальній множині U називається сукупність пар $(\mu_A(u), u)$, де $\mu_A(u)$ – ступінь належності елемента до нечіткої множини \tilde{A} . Ступінь належності – це число з діапазону $[0, 1]$. Чим вище ступінь належності, тим більшою мірою елемент універсальної множини відповідає властивостям нечіткої множини [43].

Функцією належності (membership function) називається функція, яка дозволяє обчислити ступінь належності довільного елемента універсальної множини до нечіткої множини.

Якщо універсальна множина є кінцевою $U = \{u_1, u_2, \dots, u_n\}$, тоді нечітка множина записується у вигляді:

$$\tilde{A} = \sum_{i=1}^n \mu_{\tilde{A}}(u_i)/u_i \text{ або } \tilde{A} = \mu_{\tilde{A}}(u_1)/u_1, \mu_{\tilde{A}}(u_2)/u_2, \dots, \mu_{\tilde{A}}(u_n)/u_n.$$

Лінгвістичною змінною (linguistic variable) називається змінна, значеннями якої можуть бути слова або словосполучення деякого природної або штучної мови. Терм-множиною (term set) називається множина всіх можливих

значень лінгвістичної змінної. Термом (therm) називається будь-який елемент терм-множини. В теорії нечітких множин терм формалізується нечіткою множиною за допомогою функції належності.

Лінгвістична змінна – це змінна, значення якої визначається набором вербальних (словесних) характеристик деякої властивості. Лінгвістична змінна це багато нечітких змінних, вона використовується для того щоб дати словесний опис деякому нечіткому числу, отриманому в результаті деякими операціями.

Значення лінгвістичної змінної визначається через так звані нечіткі множини, які в свою чергу визначені на деякому базовому наборі чи базовій шкалі, яка має розмірність [44].

Звичайну підмножину можна було б ототожнити з його характеристичної функцією. Цього математично неможливо зробити, оскільки для завдання функції необхідно спочатку поставити багато. Нечітку підмножину з формальної точки зору можна ототожнити з його функцією належності. Однак термін «нечітка підмножина» переважає при побудові математичних моделей реальних явищ.

Методи нечітких множин дозволяють вибирати параметри генетичних алгоритмів, а також коефіцієнти, які визначають швидкість навчання нейронних мереж [45].

Нечіткі множини визначаються через функцію належності, тобто нечіткі продукційні правила не містять будь-яких не роз'яснень. У випадку нечіткої логіки, можна отримати необмежену кількість операцій, тому не виникає потреби записати всі операції за допомогою деякого числа базових операцій, враховуючи це, необхідно обмежитися найбільш важливими операціями.

Задача продукційних правил полягає в формуванні таких нечітких правил, щоб побудованих на їх основі модуль керування при отриманні вхідних сигналів генерував коректні вихідні сигнали. Тому, нам необхідно поділити простір вхідних і вихідних сигналів на області та створити відповідні їм функції належності. Побудувати нечіткі правила на основі навчальної вибірки, створити

таблицю для запису бази правил та таблицю степенів істинності правил. Приписати кожному правилу степінь істинності та сформулювати відповідні правила, створити базу нечітких правил [46].

Практичне використання теорії нечітких множин пропонує використання функції належності, якою описуються лінгвістичні змінні «низький», «середній», «високий» і тому подібні. Задача побудови функції належності формується наступним чином. На відміну від імовірнісної міри, яка є оцінкою стохастичною невизначеності, що має справу з неоднозначністю настання деякої події в різні моменти часу, нечітка міра є чисельною оцінкою лінгвістичної невизначеності, пов'язаної з неоднозначністю і розпливчатістю категорій людського мислення.

Найпростіші функції належності трикутна і трапецієвидна формується з використанням кусково-лінійної апроксимації. Трапецієвидна функція належності є узагальненням трикутної, вона дозволяє задавати ядро нечіткої множини у вигляді інтервалу. У разі трапецієподібної функції належності можлива наступна зручна інтерпретація: ядро нечіткої множини – оптимістична оцінка; носій нечіткої множини – песимістична оцінка [47].

Дві функції належності – симетрична гаусова (gaussmf) і двостороння гаусова (gaussmf) формується з використанням гаусового розподілу. Гаусові функції дозволяє задавати асиметричні функції належності. Узагальнена дзвоноподібна функція належності за своєю формою схожа на гаусові. Ці функції належності часто використовуються в нечітких системах, так як на всій області визначення вони є гладкими і приймають ненульові значення [48].

В середовищі Fuzzy Logic Toolbox функції належності поділяються за типовою формою (зображено на рисунку 2.1):

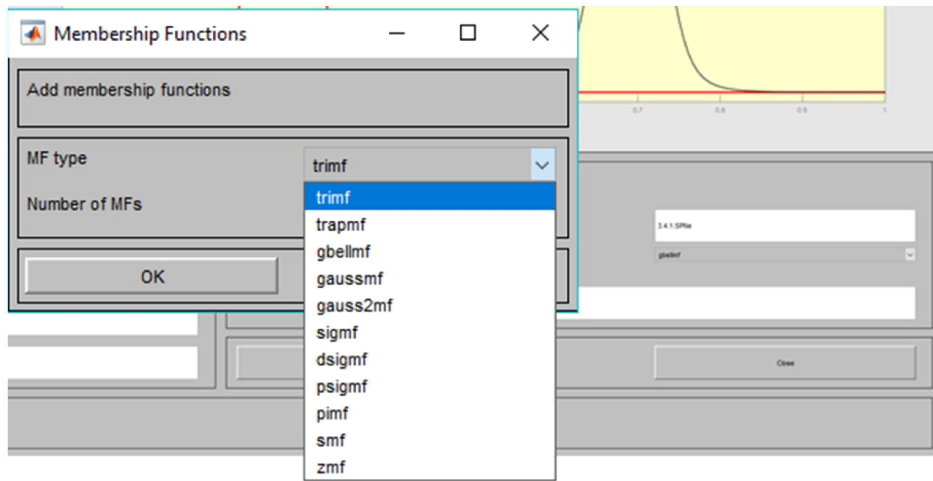


Рисунок 2.1 – Класифікація функцій належності

- 1) Трикутна функція належності, яка зображена на рисунку 2.2 (а);
- 2) трапецієподібна, наведена на рисунку 2.2 (б);
- 3) дзвоноподібної форми, рисунок 2.2 (в);
- 4) гаусову симетричну, представлено на рисунку 2.2 (г);

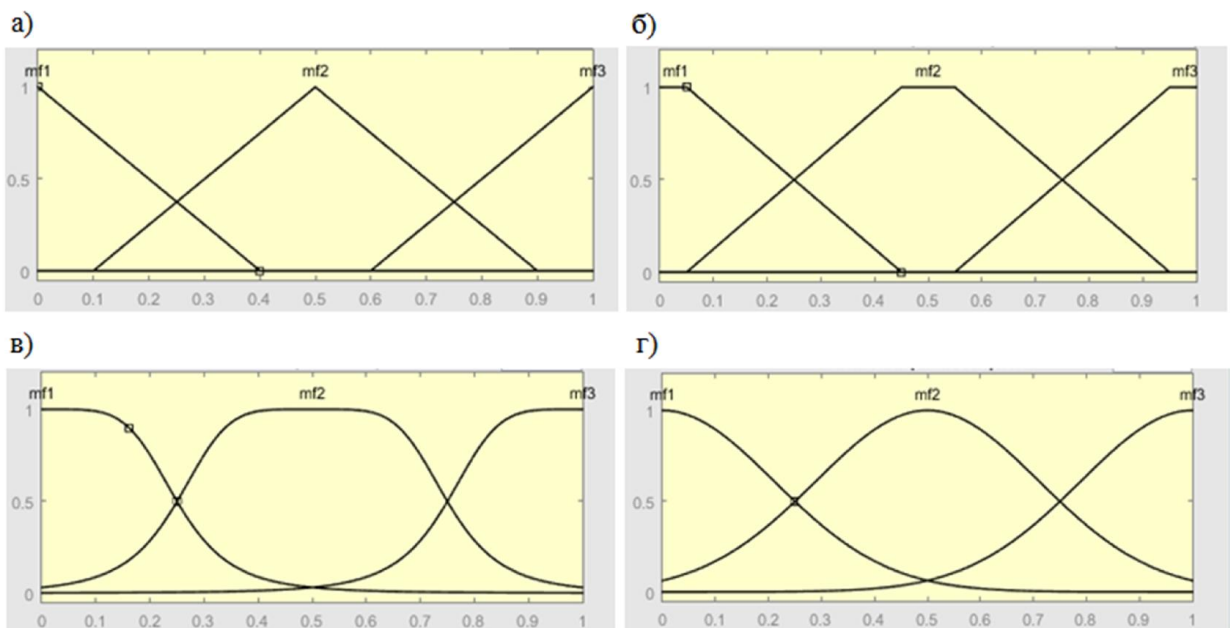


Рисунок 2.2- Функції належності. а) трикутна, б) трапецієподібна, в) дзвоноподібної форми, г) гаусова симетрична

- 5) гаусова двостороння, зображена на рисунку 2.3 (а);
- б) сигмоїдна, наведена на рисунку 2.3 (б);
- в) двостороння сигмоїдна, рисунок 2.3 (в);
- 7) S-функція належності, представлена на рисунку 2.3 (г);

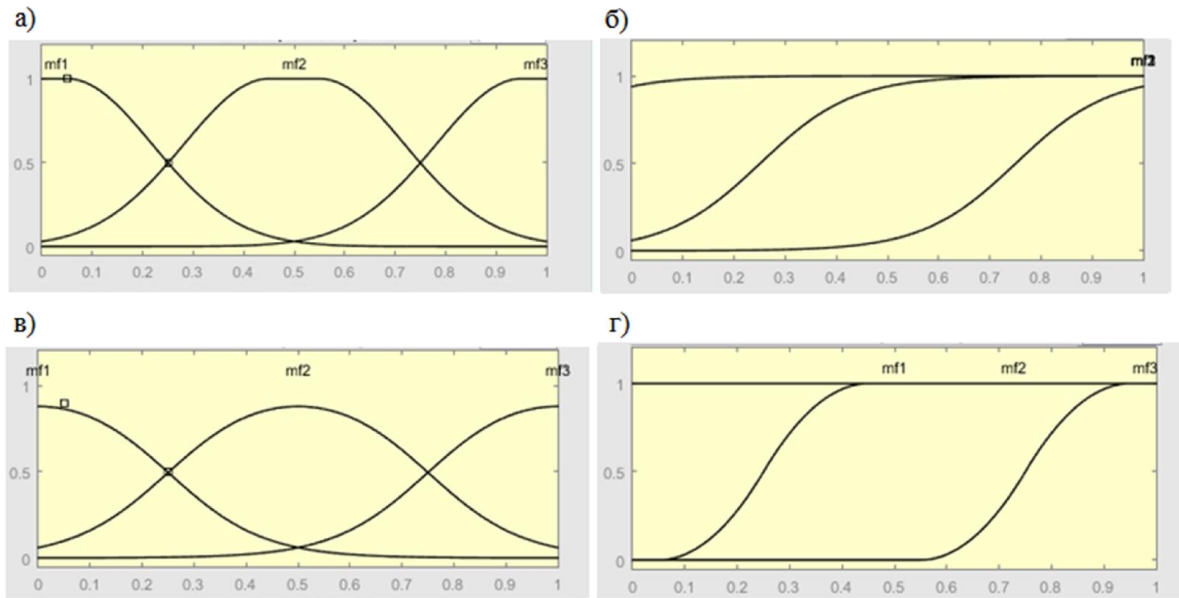


Рисунок 2.3- Функції належності. а) гаусова двостороння, б) сигмоїдна, в) двостороння сигмоїдна, г) S-функція

- 8) Z-функція належності, наведена на рисунку 2.4 (а);
- 9) P-подібна функція належності, зображена на рисунку 2.4 (б).

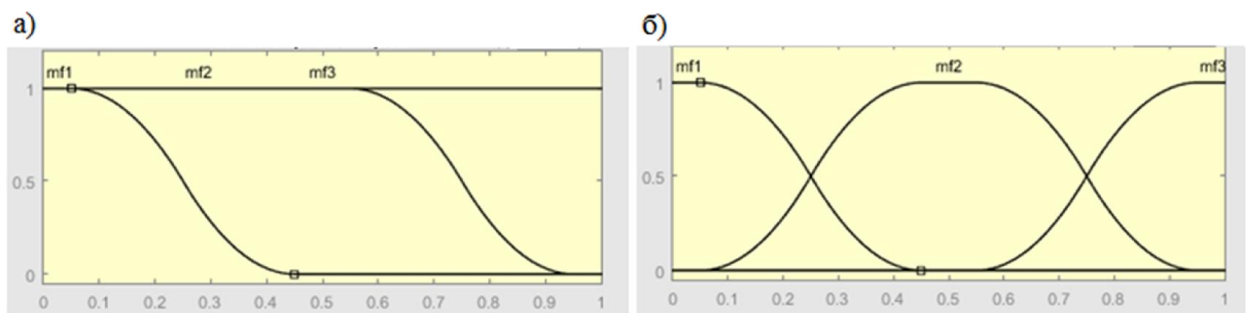


Рисунок 2.4 – Функції належності. а) Z-функція, б) P-подібна функція

Функції належності засновані на використанні сигмоїдної кривої, дозволяють формувати функції належності, значення яких починаючи з деякого значення аргументу і до + (-) рівні 1. Такі функції зручні для завдання лінгвістичних термів типу «великий» або «малий» терм-множини «Розмір клітини», зображено на рисунку 2.5 (а) та «Форма клітини» (б) [49].

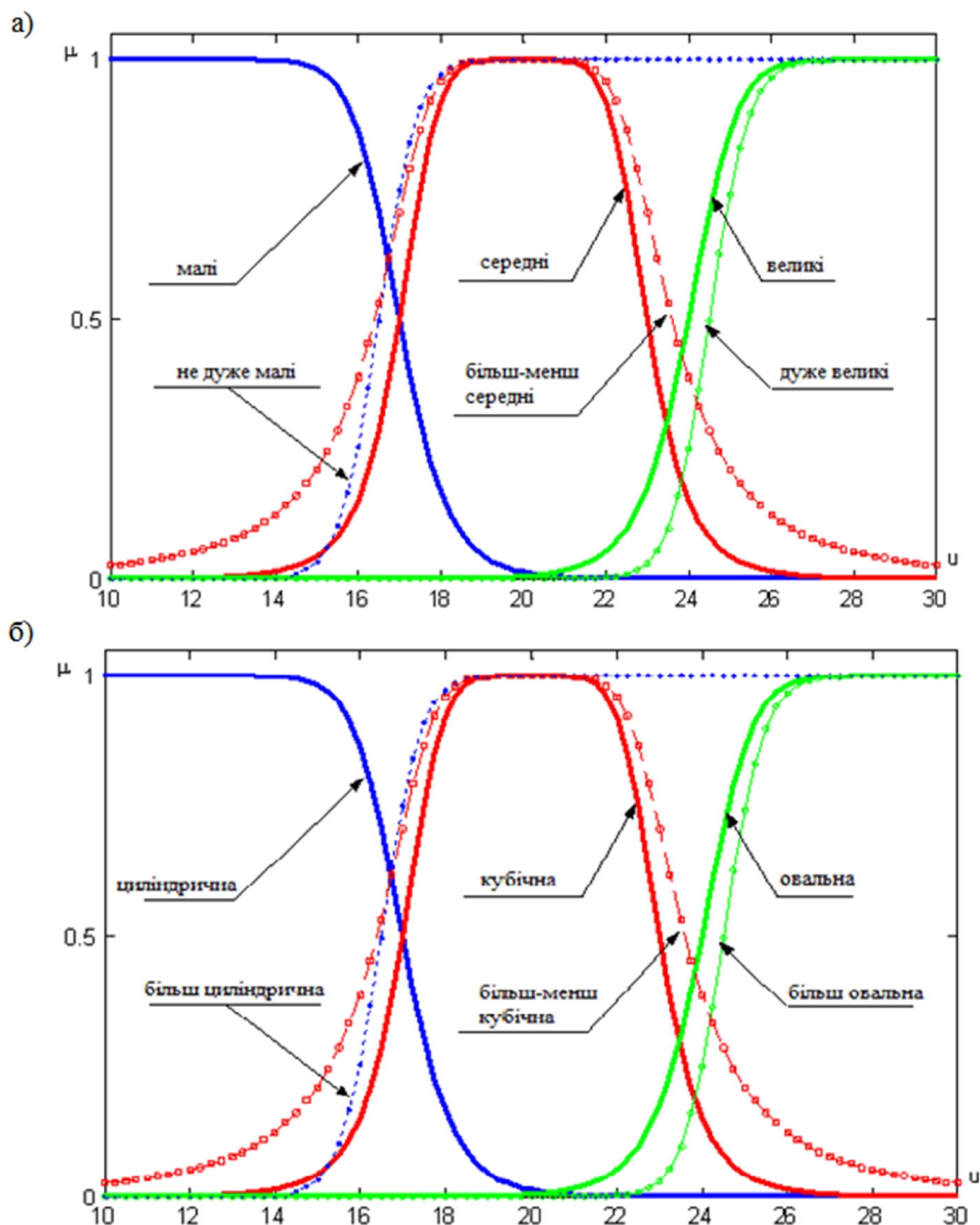


Рисунок 2.5 – Функції належності для терм-множини. а) «Розмір клітини», б) «Форма клітини»

Функції належності засновані на використанні сигмоїдної кривої також можна застосувати для значень терм-множини «Форма клітини» та матимуть вигляд, наведений на рисунку 2.7.

В якості експериментальних досліджень використано тестову навчальну вибірку цитологічних та гістологічних зображень [3]. Вхідними змінними є геометричні ознаки даних зображень, а саме розмір клітини та її форма. Змінна “розмір клітини” включає такі елементи терм-множини: малі (small), середні (medium) та великі (large). Змінна “форма клітини” включає елементи терм-множини: циліндрична (cylindrical), кубічна (cubic), овальна (oval).

Таблиця 2.1 – Ознаки мікрооб’єктів

Ознака	Терм-множина		
	малі (small)	середні (medium)	великі (large)
Розмір клітин	малі (small)	середні (medium)	великі (large)
Значення площі, мін.	12573	25791	62514
Значення площі, макс.	48748	64952	96812
Форма клітин	циліндрична (cylindrical)	кубічна (cubic)	овальна (oval)
Значення округлості, мін.	0,24	0,48	0,63
Значення округлості, макс.	0,58	0,74	0,95

Функція належності базується на отриманих експертом знаннях з мікрооб’єктів та їх числових ознаках. Після проведених експериментальних досліджень виміру розмірів та форми нормальних клітин та одного із диспластичних процесів, а саме проліферативна мастопатія, отримано числові значення, які наведені в таблиці 2.1 [1].

Вихідна змінна формується з висновків експерта на основі вхідних змінних про наявність диспластичного процесу та складається із двох нечітких множин, а саме: проліферативна мастопатія (proliferative breast disease) та нормальна тканина (normal).

Експериментальні дослідження проведено за допомогою програмного засобу Fuzzy Logic Toolbox комплексу Matlab. Результати роботи наведено на рисунках 2.8 – 2.10.

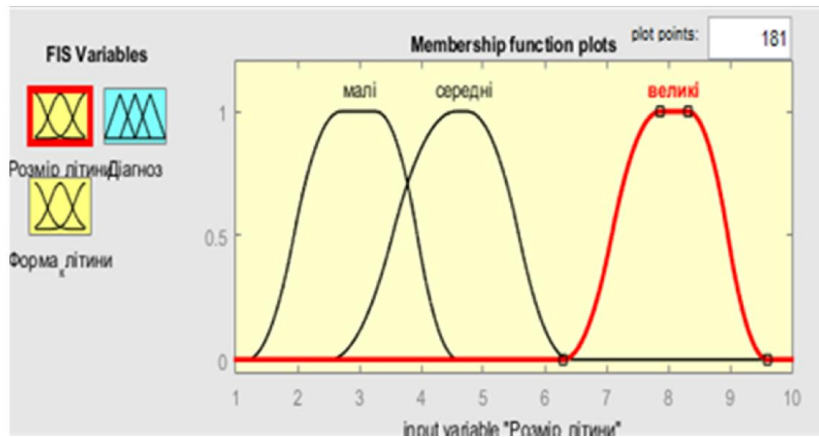


Рисунок 2.8 – Функція належності «Розмір клітини»

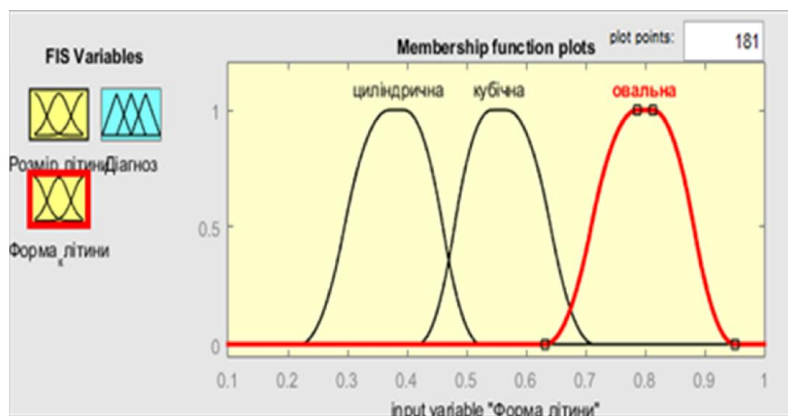


Рисунок 2.9 – Функція належності «Форма клітини»

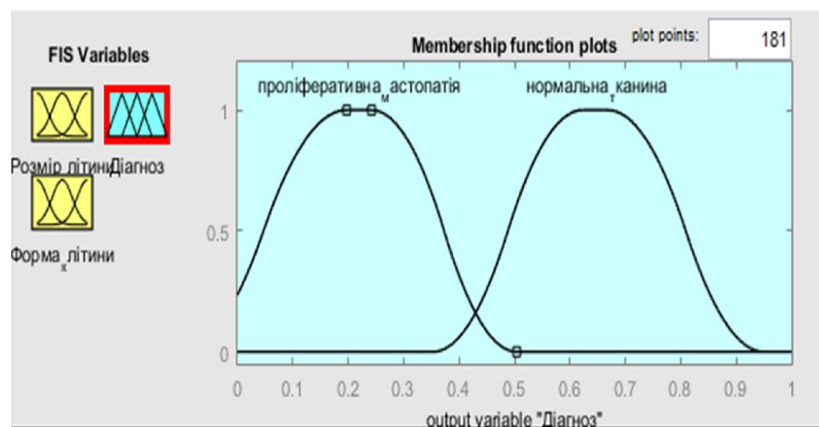


Рисунок 2.10 – Функція належності «Діагноз»

Поліноміальна апроксимація графічно відображається подібно до функції сигмоїдної кривої.

Перевагами застосування нечітких систем для побудови продукційних правил є наступними:

1) можливість оперувати вхідними даними, заданими нечітко: наприклад, що безупинно змінюються в часі значення (динамічні задачі), значення, що неможливо задати однозначно;

2) можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями «більшість», «можливе», переважно» і т.д.;

3) можливість проведення якісних оцінок як вхідних даних, так і виведених результатів: операції відбуваються не тільки власне значеннями даних, але їхнім ступенем вірогідності і її розподілом;

4) можливість проведення швидкого моделювання складних динамічних систем і їхній порівняльний аналіз із заданим ступенем точності: оперуючи принципами поведінки системи, описаними fuzzy-методами, не витрачається багато часу на з'ясування точних значень змінних і складання рівнянь, що їх описують, можлива оцінка різні варіантів вихідних значень [50].

2.3 Алгоритми фазифікації та дефазифікації

Етапи нечіткого виводу можуть бути реалізовані неоднозначним чином, оскільки включають в себе окремі параметри, які повинні бути фіксовані або специфіковані. Тим самим вибір конкретних варіантів параметрів кожного з етапів визначає певний алгоритм, який в повному обсязі реалізує нечіткий висновок в системах правил нечітких продукцій [51].

Модель нечіткого виведення зображено на рисунку 2.11.

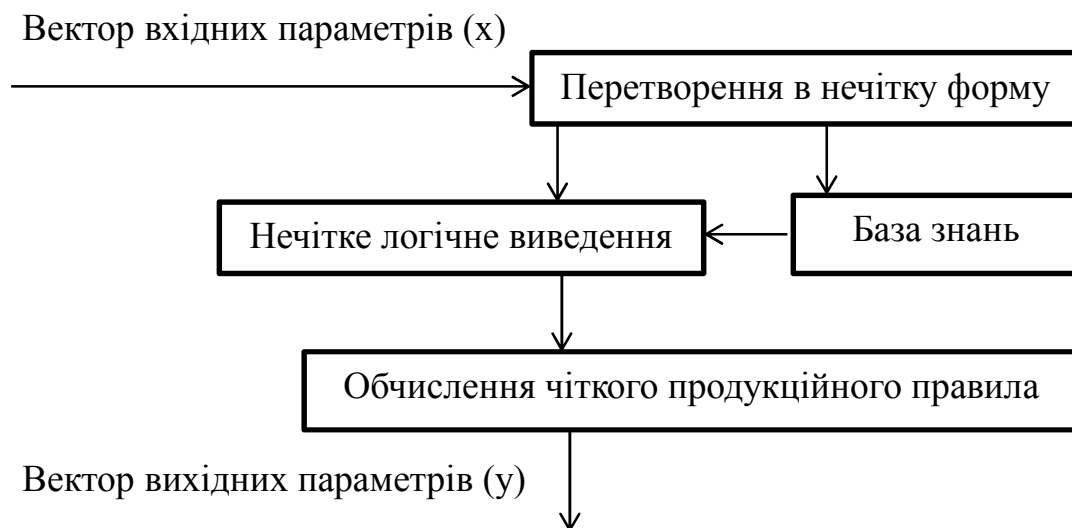


Рисунок 2.11 – Модель нечіткого виведення

Нечіткий логічний вивід відбувається через наступні етапи:

1) Нечіткість (введення нечіткості, фазифікація). Функції належності, які визначені на вхідних змінних застосовуються до їх фактичних значень для визначення істинності кожної передумови кожного правила.

2) Логічний вивід. Визначення значення істинності для передумови кожного правила використовується для висновків кожного з правил. Це приводить до однієї нечіткої підмножини, яка буде призначена кожній змінній виводу для кожного правила.

3) Композиція . Нечіткі підмножини, призначені для кожної змінної виводу, об'єднуються разом, щоб сформувати одну нечітку підмножину для кожної змінної виводу.

4) Дефазифікація – використовується, коли необхідно перетворити нечітку множину виводів в чітке число. В теорії нечітких множин процедура дефазифікації аналогічна процедурі знаходження характеристик випадкових величин (математичне очікування, моди, медіани) в теорії ймовірності. Найпростішим способом виконання процедури дефазифікації є вибір чіткого числа, який відповідає максимуму функції належності [51].

Однак придатність цього способу обмежується лише однокстремальними

функціями належності. Для багатоекстремальних функцій належності в Fuzzy Logic Toolbox запрограмовані такі методи дефазифікації:

- 1) Centroid – центр ваги;
- 2) Bisector – медіана;
- 3) LOM (Largest Of Maximums) – найбільший з максимумів;
- 4) SOM (Smallest Of Maximums) – найменший з максимумів;
- 5) Mom (Mean Of Maximums) – центр максимумів.

До теперішнього часу в середовищі Fuzzy Logic Toolbox програмного комплексу Matlab запропоновано кілька алгоритмів нечіткого висновку, а саме алгоритми Mamdani та Sugeno [52].

За своєю суттю алгоритм Мамдані породжує розглянуті вище етапи, оскільки найбільшою мірою відповідає їх параметрами.

Формально алгоритм Мамдані може бути визначений таким чином.

- 1) Формування бази правил систем нечіткого виводу.
- 2) Фазифікації вхідних змінних.
- 3) Агрегація підумови в нечітких правилах продукцій. Для знаходження степені істинності умов кожного з правил нечітких продукцій використовуються парні нечіткі логічні операції. Ті правила, ступінь істинності умов яких відмінна від нуля, вважаються активними і використовуються для подальших розрахунків.
- 4) Активізація підзаключень в нечітких правилах продукцій. Здійснюється за формулою, при цьому для скорочення часу виведення враховуються тільки активні правила нечітких продукцій.
- 5) Акумуляція висновків нечітких правил продукцій. Здійснюється за формулою для об'єднання нечітких множин, відповідних термам підзаключень, що відносяться до одних і тих же вихідних лінгвістичних змінних.
- 6) Дефазифікації вихідних змінних. Традиційно використовується метод центру ваги в формі або метод центру площі [54].

Формування бази правил систем нечіткого виводу. У базі правил

використовуються тільки правила нечітких продукцій у формі:

Правило: якщо « $\beta_1 \in \alpha$ » I « $\beta_2 \in \alpha$ », то « $w = \varepsilon_1 \cdot a_1 + \varepsilon_2 \cdot a_2$ »,

де $\varepsilon_1, \varepsilon_2$ – деякі вагові коефіцієнти. При цьому значення вихідної змінної w у висновку визначається як деяке дійсне число.

2) Фазифікації вхідних змінних. Особливості фазифікації збігаються з розглянутих вище при описі даного етапу.

3) Агрегація підумови в нечітких правилах продукцій. Для знаходження степені істинності умов всіх правил нечітких продукцій, як правило, використовується логічна операція \min -кон'юнкції. Ті правила, ступінь істинності умов яких відмінна від нуля, вважаються активними і використовуються для подальших розрахунків.

4) Активізація підзаключень в нечітких правилах продукцій, знаходяться значення ступенів істинності всіх висновків правил нечітких продукцій, здійснюється розрахунок звичайних (не нечітких) значень вихідних змінних кожного правила. Це виконується за допомогою формули для висновку, в яку замість a_1 і a_2 підставляють значення вхідних змінних до етапу фазифікації. Тим самим визначаються множина значень $C = \{c_1, c_2, \dots, c_n\}$ і множина значень вихідних змінних $W = \{w_1, w_2, \dots, w_n\}$, де n – загальна кількість правил в базі правил.

5) Акумуляція висновків нечітких правил продукцій фактично відсутня, оскільки розрахунки здійснюються із звичайними дійсними числами.

6) Дефазифікації вихідних змінних [55].

2.4 Узагальнений алгоритм побудови нечітких правил

Нечітка система на основі продукційних правил є найбільш розповсюдженою при моделюванні складних систем, тому що вона використовує лінгвістичні змінні. Лінгвістичні змінні можуть бути природним чином представлені в нечітких множинах та у ролі логічних зв'язків цих

множин. Нечіткий рівень розуміння і опису складної системи виражається у вигляді набору обмежень на виході за рахунок певних умов вводу. Обмеження, як правило, моделюються нечіткими множинами та зв'язками типу «and», «or», «then» [56].

Системи нечіткого виводу призначені для перетворення значень вхідних змінних у вихідні змінні на основі використання нечітких правил продукції.

Узагальнений алгоритм представимо у вигляді таких кроків:

- 1) формування бази правил;
- 2) фазифікація вхідних змінних;
- 3) агрегування підумови в нечітких правилах продукції;
- 4) активізація або композиція підзаклучень;
- 5) акумулювання висновків в нечітких правилах продукції;
- 6) дефазифікація вхідних змінних [58].

Оскільки зазвичай правил багато, їх об'єднують в множину, яка називається базою правил. Найчастіше база правил представляється у вигляді:

Правило 1. ЯКЩО «Умова 1», ТО «ВИВІД 1»

Правило 2. ЯКЩО «Умова 2», ТО «ВИВІД 2»

...

Правило n. ЯКЩО «Умова n», ТО «Вивід n».

На основі проведених результатів отримано правила передракових станів:

IF розмір клітин IS малі (small) AND форма клітин IS овальна (oval)

THEN діагноз IS нормальна тканина (normal).

IF розмір клітин IS середні (medium) OR розмір клітин IS великі (large)
AND форма клітин IS циліндрична (cylindrical) OR форма клітин IS кубічна
(cubic) THEN діагноз IS проліферативна мастопатія (proliferative breast disease).

Фазифікацією називається процес або процедура знаходження значень функцій належності нечітких множин (термів) на основі звичайних («чітких») вихідних даних. Метою фазифікації є встановлення відповідності між

конкретним значенням вхідної змінної і значенням функції належності відповідно її теорема вхідної лінгвістичної змінної [1].

Агретування являє собою процедуру визначення ступеня істинності умов по кожному з правил системи нечіткого виводу, наведено на рисунку 2.12.

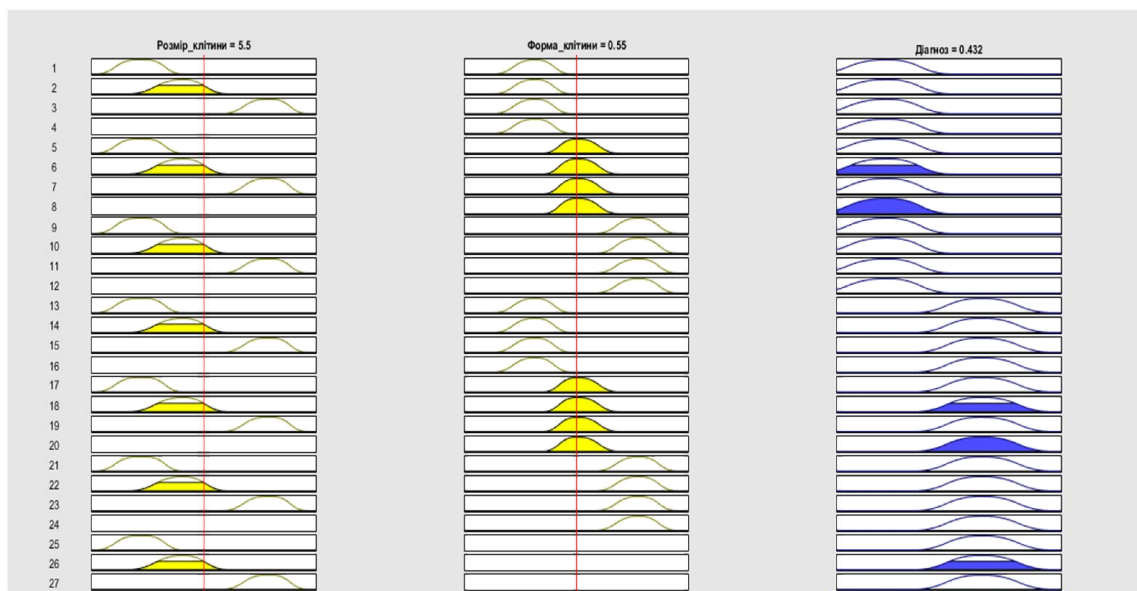


Рисунок 2.12 – Нечіткі продукційні правила на основі вхідних змінних

Активізація являє собою процедуру або процес знаходження ступеня істинності кожного з підзаключення правил нечіткої продукції. Знаходяться всі значення ступенів істинності підзаключення для кожного з правил, що входить до бази правил. Після знаходження множини визначається функція належності кожного з підзаключення для розглянутих вхідних лінгвістичних змінних [1].

Акумуляція (акумуляування) являє собою процедуру або процес об'єднання всіх ступенів істинності висновків для знаходження функції належності для кожної з вихідних лінгвістичних змінних, зображено на рисунку 2.13.

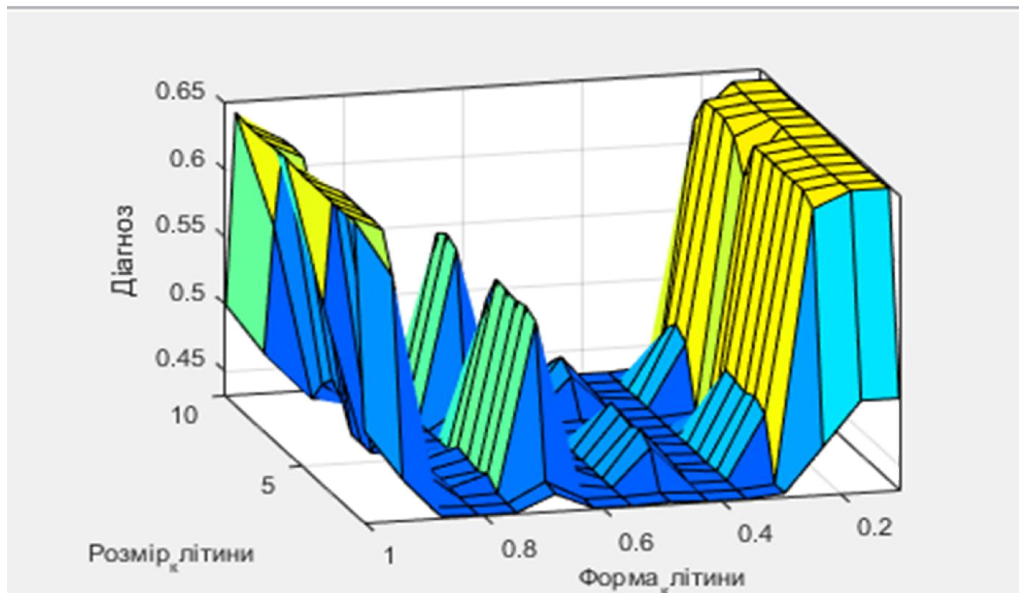


Рисунок 2.13 – Поверхня значень залежності виходу нечіткої системи від вхідних значень «розмір клітини» і «форма клітини»

Дефазифікація являє собою процедуру або процес знаходження звичайного «чіткого» значення для кожної з вихідних лінгвістичних змінних. Дефазифікації виробляють з використанням деяких аналітичних виразів, іменованих методами дефазифікації [57].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ НЕЧІТКИХ ПРОДУКЦІЙНИХ ПРАВИЛ

3.1 Програмно-апаратний засіб

Для побудови алгоритму функцій належності необхідно правильно підібрати програмно-апаратний комплекс, який забезпечить швидку та безперебійну роботу системи. Характеристики апаратного забезпечення наведено в таблиці 3.1.

Таблиця 3.1 – Характеристики апаратного забезпечення

Система	Характеристики
Виробник	Dell Inc.
Модель	Latitude
Процесор	Intel(R) Core(TM) i5-2520M CPU 2,50GHz
Установлена оперативна пам'ять	4,00 ГБ
Тип системи	64-розрядна ОС на базі процесора x64
Операційна система	Windows 10 Pro

Характеристики програмного комплексу Matlab на моделі даного апаратного забезпечення наведено в таблиці 3.2.

Таблиця 3.2 – Характеристики програмного забезпечення

Програмне забезпечення	Характеристики
Розробник	The MathWorks
Версія	R2015a (8.5.0.197613)
Тип ПЗ	64-bit (win64)
Зчитувані формати файлів	Файл MATLAB-M
Створювані формати файлів	Файл MATLAB-M

Пакет Fuzzy Logic Toolbox або пакет нечіткої логіки - це сукупність прикладних програм, що відносяться до теорії розмитих або нечітких множин, дозволяє конструювати так звані нечіткі експертні або керуючі системи [58].

Основні можливості пакету:

- 1) побудова систем нечіткого виводу (регуляторів, апроксиматорів залежностей);
- 2) побудова адаптивних нечітких систем (гібридних нейронних мереж);
- 3) інтерактивне динамічне моделювання в Simulink.

Пакет дозволяє роботу:

- 1) в режимі графічного інтерфейсу;
- 2) в режимі командного рядка;
- 3) з використанням блоків і прикладів пакета Simulink.

До складу програмних засобів Fuzzy Logic Toolbox входять наступні основні програми, що дозволяють працювати в режимі графічного інтерфейсу:

- 1) редактор нечіткої системи виведення Fuzzy Inference System Editor (FIS Editor або FIS-редактор) разом з допоміжними програмами, а саме: редактором функцій належності (Membership Function Editor), редактором правил (Rule Editor), переглядачем правил (Rule Viewer) і переглядачем поверхні відгуку (Surface Viewer);
- 2) редактор гібридних систем (ANFIS Editor, ANFIS-редактор);
- 3) програма перебування центрів кластерів (програма Clustering - кластеризація).

Fuzzy Logic Toolbox володіє простим і добре продуманим інтерфейсом, що дозволяє легко проектувати і діагностувати нечіткі моделі. Забезпечується підтримка сучасних методів нечіткої кластеризації і адаптивні нечіткі нейронні мережі. Графічні засоби Fuzzy Logic Toolbox дозволяють інтерактивно відстежувати особливості поведінки системи.

Набір інструментів дозволяє моделювати складні системи поведінки, використовуючи прості логічні правила, а потім реалізувати ці правила в

нечіткої системи логічного висновку. Fuzzy Logic Toolbox можна використовувати його в якості автономного нечіткого логічного висновку.

Функції Matlab, які можуть бути використані для виклику відповідних графічних засобів, представлені в таблиці 3.3.

Таблиця 3.3 – Функції графічного інтерфейсу

№	Назва функції	Призначення
1.	Anfisedit	Редактор гібридних мереж ANFIS
2.	Findcluster	Програма нечіткої кластеризації
3.	Fuzzy	Редактор системи нечіткого виведення FIS
4.	Mfedit	Редактор функцій належності
5.	Ruleedit	Редактор правил нечіткого виводу
6.	Ruleview	Програма перегляду правил і діаграми нечіткого виведення
7.	Surfview	Програма перегляду поверхні нечіткого виведення

Редактор систем нечіткого виводу FIS (або просто редактор FIS) є основним засобом, який використовується для створення або редагування систем нечіткого виводу в графічному режимі.

Ця функція надає користувачеві можливість задавати і редагувати на високому рівні якості системи нечіткого виведення, такі, як число вхідних і вихідних змінних, тип системи нечіткого висновку, що використовується метод дефаззифікації та інших.

Набір даних програм надає користувачеві максимальні зручності для створення, редагування та використання різних систем нечіткого виводу.

Редактор FIS може бути відкритий за допомогою введення функції «fuzzy» або «fuzzy ('FISfile')» в командному рядку. Fuzzy Logic - редактор нечіткої системи виведення (Fuzzy Inference System Editor, FIS Editor, FIS-редактор) [59].

Якщо функція «fuzzy» викликається без аргументів, то редактор FIS викликається для новостворюваної системи нечіткого виведення з назвою «Untitled». За замовчуванням також задається ряд таких параметрів, таких як тип системи нечіткого виведення (Мамдані), нечіткі логічні операції, методи імплікації, агрегування, дефазифікації і деякі інші.

У верхній частині робочого інтерфейсу редактора FIS відображається діаграма, що представляє у візуальній формі входи і виходи системи нечіткого виведення, в центрі яких знаходиться редактор нечітких правил «Rule editor». Натискання маніпулятором типу миша на прямокутнику з зображенням входу або виходу виділяє відповідну змінну і робить її поточною. Зображення поточної змінної при цьому виділяється червоним кольором.

Подвійне натискання на маніпулятор типу миш на прямокутнику з зображенням вхідних або вихідних змінних викликає редактор функцій належності з завантаженою в нього відповідною змінною. Подвійне натискання маніпулятора типу миш на зображенні процесора нечітких правил викликає редактор правил для відповідної системи нечіткого виведення.

Вигляд початкової сторінки зображено на рисунку 3.1. Головне меню редактора містить позиції:

1) File - робота з файлами моделей містить в собі операції створення, імпорту як з робочого середовища, так само і з диску, експорту в робоче середовище, збереження в робоче середовище чи на диск, зчитування і друк;

2) Edit - операції редагування включає операції додавання, відміни і виключення вхідних і вихідних змінних, а також містить редактор для побудови нечітких продукційних правил;

3) View - перехід до додаткового інструментарію, за допомогою якого можна переглянути продукційні правила та поверхню значень нечіткого виводу.

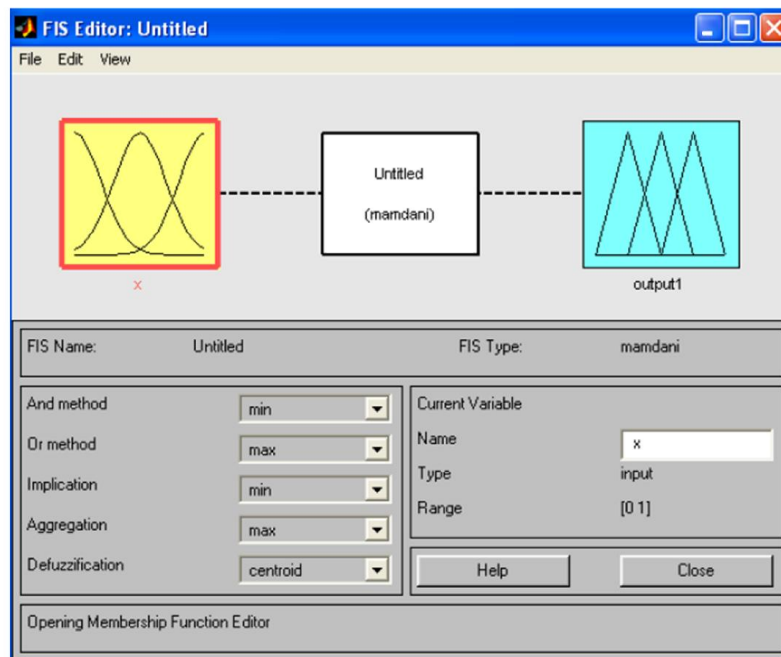


Рисунок 3.1 – Вигляд початкового екрану роботи FIS-редактору

Алгоритм дій побудови нечітких продукційних правил в FIS-редакторі матиме наступний вигляд.

1) У позиції меню File вибираємо опцію New Mamdani FIS (нова система типу Mamdani), при цьому в блоці, який відображається білим квадратом, у верхній частині вікна редактора з'явиться напис Untitled2 (mamdani).

2) Скориставшись лівою кнопкою маніпулятора типу миш на блоці input1 (вхід 1). Потім в правій частині редактора в поле під назвою Name (Ім'я), замість input1 вводимо позначення нашого аргументу, тобто x.

3) Подвійним натиском на даний перед нами відкривається вікно редактору функцій належності - Membership Function Editor (зображено на рисунку 3.2). Увійдемо в позицію меню Edit даного редактора і виберемо в ньому опцію Add MFs (Add Membership Functions - додати функцій належності). При цьому з'явиться діалогове вікно, яке зображено на рисунку 3.3, що дозволяє задати тип і кількість функцій належності. Для прикладу вибираємо гаусові функції належності (gaussmf), а їх кількість задаємо рівним п'яти. Щоб підтвердити введення інформації необхідно натиснути кнопку «ОК», після чого відбудеться повернення до вікна редактора функцій належності.

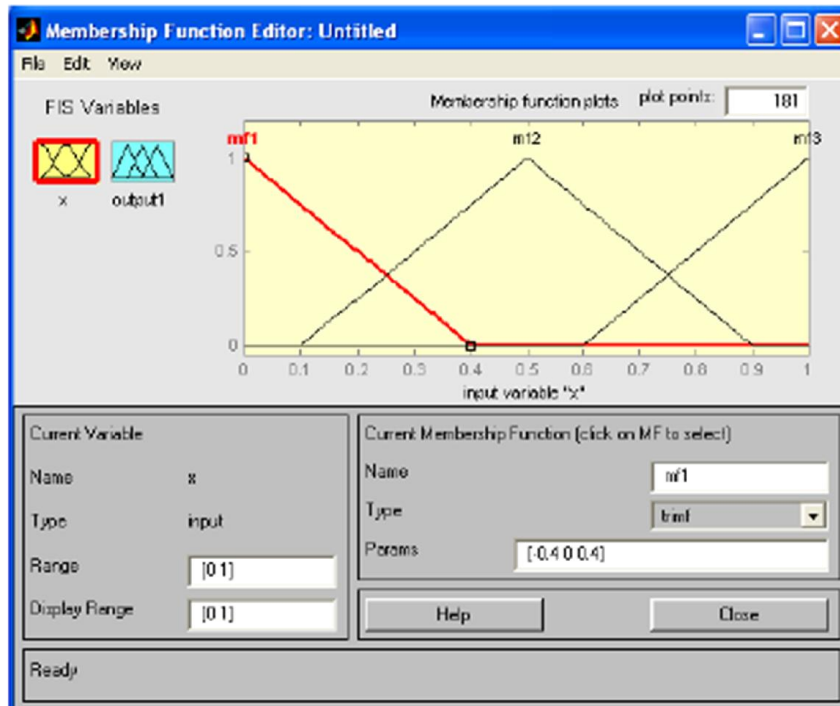


Рисунок 3.2 – Вікно редактора функцій належності

1. В полі Range встановлюється діапазон зміни x від -1 до $+1$. Потім лівою кнопкою маніпулятора типу миш натискаємо де-небудь в полі редактора.

2. Звернемося до графіків заданих нами функцій належності, зображених у верхній частині вікна редактора функцій належності. Для успішного вирішення поставленого завдання необхідно, щоб ординати максимумів цих функцій збігалися з заданими значеннями аргумента x . Для лівої, центральної та правою функцій таку умову виконано, але дві інші необхідно «посунути» уздовж осі абсцис шляхом: підводимо курсор до потрібної кривої і натискаємо лівою кнопкою маніпулятора типу миші. Крива вибирається, забарвлюючись в червоний колір, після чого за допомогою курсору її і можна посунути в потрібну сторону (більш точну установку можна провести, змінюючи числові значення в поле Params в даному випадку кожної функції належності відповідають два параметра, при цьому перший визначає розмах кривої, а другий - положення її центру).

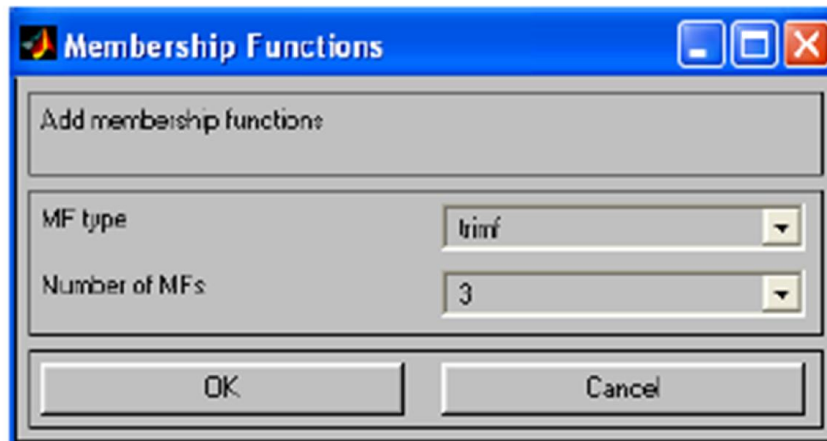


Рисунок 3.3 – Діалогове вікно для задання значень функції належності

3. Зробимо одноразове натискання лівою кнопкою маніпулятора типу миш по блоці output1 (вихід 1). У віконці Name замінимо ім'я output1 на у.

4. Двічі натиснемо по зазначеній блоку і перейдемо до програми - редактору функцій належності. У позиції меню Edit виберемо опцію Add MFs. З'являється діалогове вікно, яке дозволяє задати тепер в якості функцій належності тільки лінійні (linear) або постійні (constant), зображено на рисунку 3.3.

5. Діапазон (Range) значення змінних встановлює за замовчуванням - [0, 1], міняти не потрібно. Змінимо лише імена функції належності. Потім закриємо вікно, використовуючи кнопку Close і повертаємося у вікно FIS-редактора.

6. Двічі натиснемо лівою кнопкою маніпулятора типу миші по середньому блоці, при цьому відкриється вікно ще однієї програми - редактора правил (Rule Editor), зображено на рисунку 3.4. Введемо відповідні правила. При введенні кожного правила необхідно позначити відповідність між кожною функцією належності аргументу x і числовим значенням у і натиснемо кнопку Add rule. Введене правило з'явиться у вікні правил і буде являти собою запис:

If (x is bn) then (y is 1).

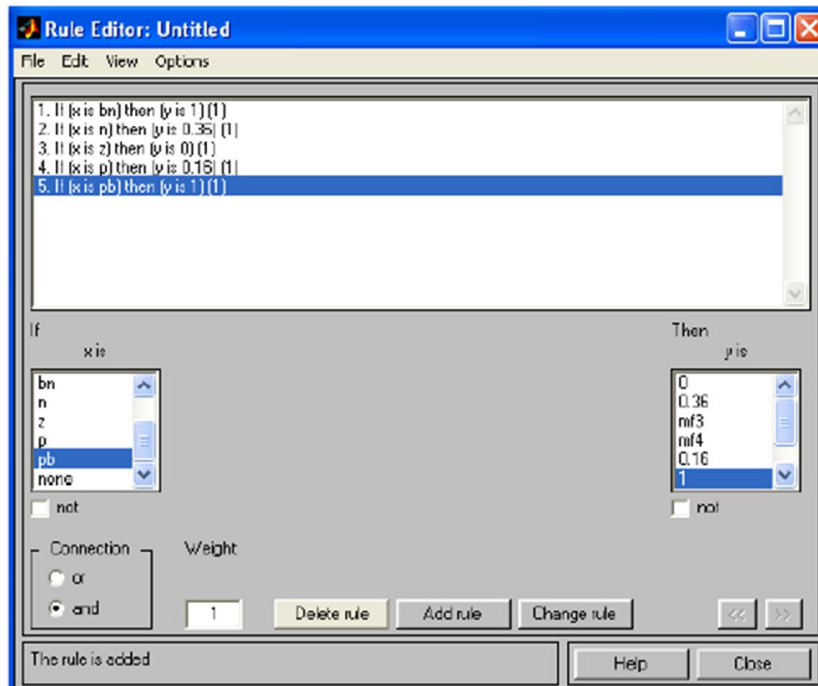


Рисунок 3.4 – Вигляд вікна редактору правил

7. Попередньо збережемо наш проект на диску, використовуючи шлях File/Save to disk as... під будь-яким іменем.

8. Виберемо позицію меню перегляд View. Як видно з випадуючого при цьому підменю, за допомогою пунктів Edit membership functions і Edit rules можна здійснити перехід до двох вище розглянутих програмами - редакторам функцій належності і правил. Перегляд правил View rules та Перегляд поверхні View surface. Вибравши пункт View rules, при цьому відкриється вікно, яке наведено на рисунку 3.5 ще однієї програми - перегляду правил Rule Viewer.

9. У правій частині вікна в графічній формі представлені функції належності аргументу ж, в лівій - змінної виходу у з поясненням механізму прийняття рішення. Червона вертикальна лінія, яка перетинає графіки в правій частині вікна, яку можна переміщати за допомогою маніпулятора типу миш, дозволяє змінювати значення змінної входу , при цьому відповідно змінює значення у в правій верхній частині вікна. Зміна аргументу шляхом переміщення червоної вертикальної лінії дуже наочно демонструє, як система визначає значення виходу.

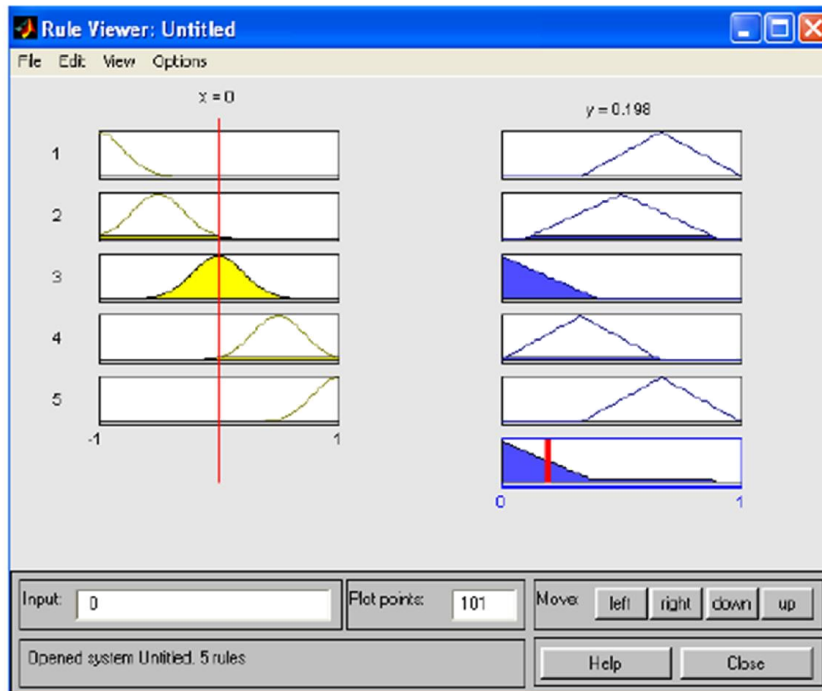


Рисунок 3.5 – Вікно перегляду правил

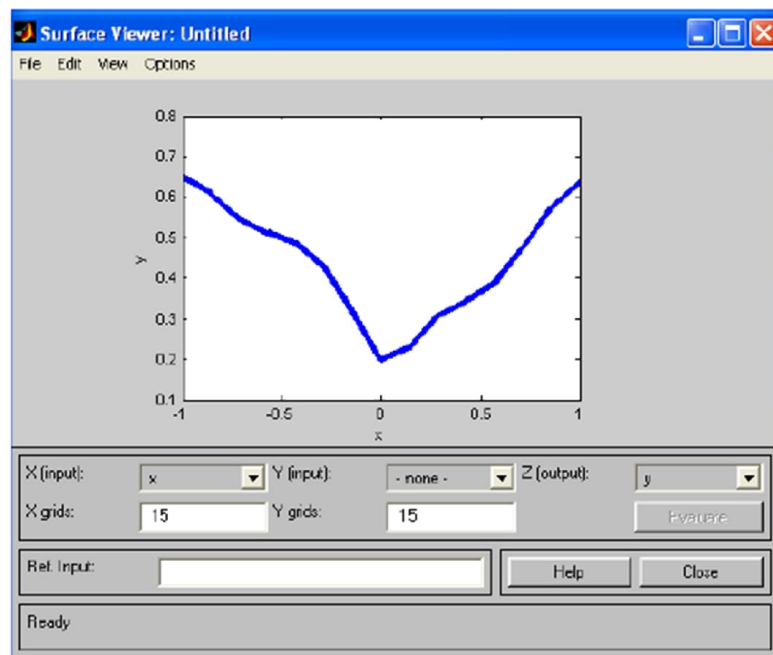


Рисунок 3.6 – Вікно перегляду поверхності

10. Закриємо вікно перегляду правил і вибором пункту меню «View», «View surface» і перейдемо до вікна перегляду поверхні відгуку (виходу), в нашому випадку – до перегляду кривої $y(x)$, яка наведена на рисунку 3.6.

3.2. Побудова функцій належності

Експертні знання використовуються для якісного опису мікрооб'єктів біомедичних зображень. Далі такі знання структуруються та записуються у відповідні класи. Для кожної такої морфологічної характеристики визначаються певні значення, які присвоюються певній ознаці. Такі ознаки виступають лінгвістичними змінними, які застосовуються для перетворення якісних ознак в кількісні ознаки для подальшого визначення їх функцій належності.

Функції належності традиційно визначають за експертною інформацією. Методом статистичної обробки експертної інформації будують функції належності, які узагальнюють думки колективу фахівців щодо розподілу елементів за певними класами та множинами.

З наведеної нижче таблиці видно, що побудова функцій належності на основі ознак мікрооб'єктів потребує згрупування та класифікації цих ознак до певних критерій. Тому експертом було запропоновано згрупувати ознаки на основі їх взаємного просторового розміщення та структурних залежностей за такими критеріями:

- 1) частина;
- 2) ознака;
- 3) стани.

Побудова функції належності нечіткої множини або нечіткого числа (нечіткої величини) має важливе практичне, тому що у більшості запропонованих випадків функція належності задається експертами на основі суб'єктивних суджень, тобто невипадково. Функція належності, у загальному випадку, є формалізованим описом ступеня присутньої нечіткості на деякій множині, яка, у свою чергу, являє собою носій нечіткої величини

Використовуючи навчальну вибірку гістологічних зображень та їхні якісні ознаки, наведені в таблиці 3.4, необхідно побудувати функції належності.

Таблиця 3.4 – Ознаки цитологічних зображень

Частина	Ознака	Стани
1.Епітелій	1.1.Сплющений апокриновий	1.1.1.Не візуалізується; 1.1.2.Добре візуалізується; 1.1.3.Слабо візуалізується
	1.2.Гіперплазія епітеліальних клітин	1.2.1.Немає; 1.2.2.Невелика кількість; 1.2.3.Переважає більшість
2.Ядра	2.1.Округлі гіперхромні	2.1.1.Немає; 2.1.2.Невелика кількість; 2.1.3.Переважає більшість
	2.2.Округлі	2.2.1.Немає; 2.2.2.Є
	2.3.Розмір ядра з інтенсивно вираженим хроматином	2.3.1.Невеликий; 2.3.2.Великий; 2.3.3.Неможливо визначити
3.Клітини	3.1.Секреторна активність	3.1.1.Інтенсивна; 3.1.2.Незначна; 3.1.3.Немає
	3.2.Розмір	3.2.1.Невеликий; 3.2.2.Великий; 3.2.3.Неможливо визначити
	3.3.Гіперхромні мономорфні	3.3.1.Немає, 3.3.2.Невелика кількість, 3.3.3.Велика кількість
	3.4.Сплющені полігональні	3.4.1.Немає; 3.4.2.Є
	3.5.Фагоцити, гістоцити	3.5.1.Немає, 3.5.2.В малій кількості, 3.5.3.В великій кількості
	3.6.Апокринова секреція	3.6.1.Немає; 3.6.2.Є
	3.7.Дистрофічні зміни	3.7.1.Немає; 3.7.2.Є

Продовження таблиці 3.3

4.Структури (тканини)	4.1.Сосочкові структури	4.1.1.Розрізнено; 4.1.2.Групами; 4.1.3.Вогнищами; 4.1.4.Багатошарові; 4.1.5.Мікроскопічно малі структури; 4.1.6.Немає
	4.2.Вузкий обідок інтенсивно пофарбованої цитоплазми	4.2.1.Немає; 4.2.2.Невелика кількість; 4.2.3.Переважає більшість
	4.3.Розростання внутрішньо-долькової сполучної тканини	4.3.1.Немає; 4.3.2.Невелика кількість; 4.3.3.Переважає більшість
	4.4.Фібробласти	4.4.1.Немає; 4.4.2.Невелика кількість; 4.4.3.Переважає більшість
	4.5.Протоки і дольки	4.5.1.Немає; 4.5.2.Вузькі; 4.5.3.Розширені
	4.6.Клітинні комплекси (ацинуси)	4.6.1.Кубічні; 4.6.2.Призматичного епітелію; 4.6.3.Немає
	4.7.Клітинні пласти	4.7.1.Немає; 4.7.2.Є
	4.8.Мікросомальні структури цитоплазми	4.8.1.Немає; 4.8.2.Гомогенна, 4.8.3.Мікрозерниста
	4.9.Сосочкоподібні комплекси	4.9.1.Немає, 4.9.2.Є
	4.10.Однотипні мілкі трубочки	4.10.1.Немає; 4.10.2.Є
	4.11.Строма	4.11.1.Слабо розвинена; 4.11.2.Сильно розвинена, 4.11.3.Немає
	4.12.Гіалінозстроми	4.12.1.Немає; 4.12.2.Є

Першим кроком для побудови функцій належності буде відкриття середовища Matlab, а в ньому необхідно запусити редактор виводу нечіткої логіки – Fuzzy Logic Toolbox, зображений на рисунку 3.7.

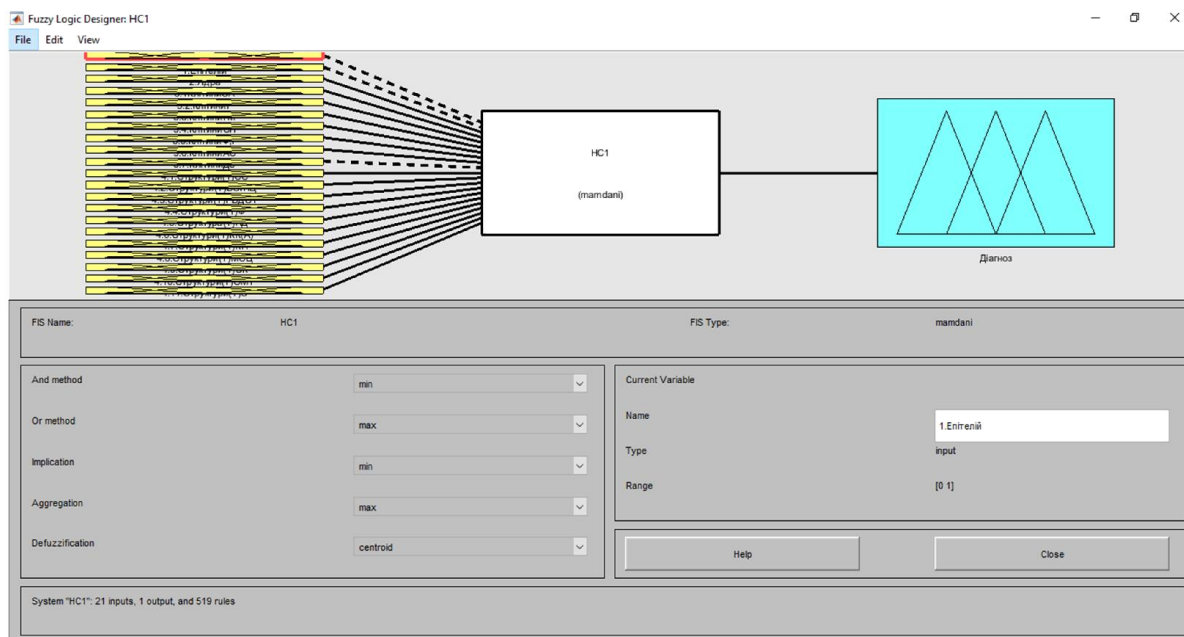


Рисунок 3.7 – Редактор «Fuzzy Logic Toolbox»

Якісні ознаки, які містяться в таблиці 3.1 виступають вхідними даними, а вихідними – правильно поставлений діагноз. Оскільки дана вибірка біомедичних зображень містить понад 20 вхідних значень, їх можна додати (рисунок 3.8.) вибираючи пункт меню «Edit», далі «Add variable» та вибираємо пункт «Input».

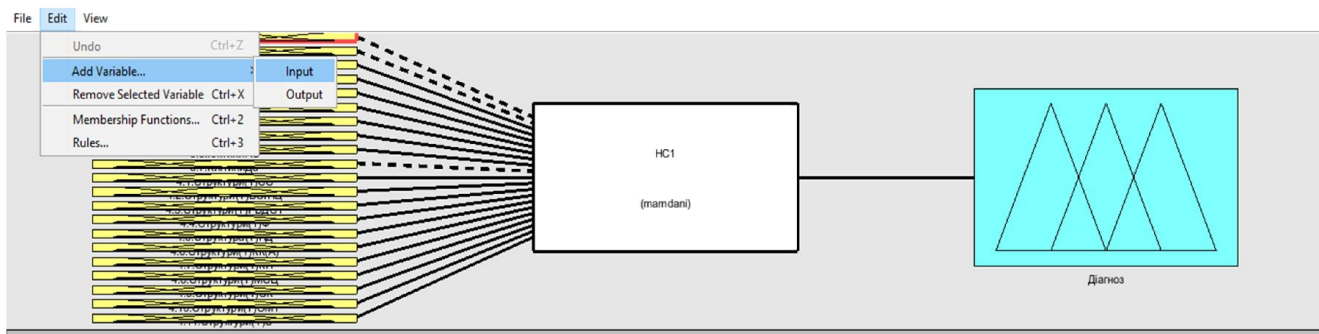


Рисунок 3.8 – Додавання вхідних даних в редакторі «Fuzzy Logic Toolbox»

В кінцевому результаті матимемо вигляд вхідних даних, які зображено на рисунку 3.9.

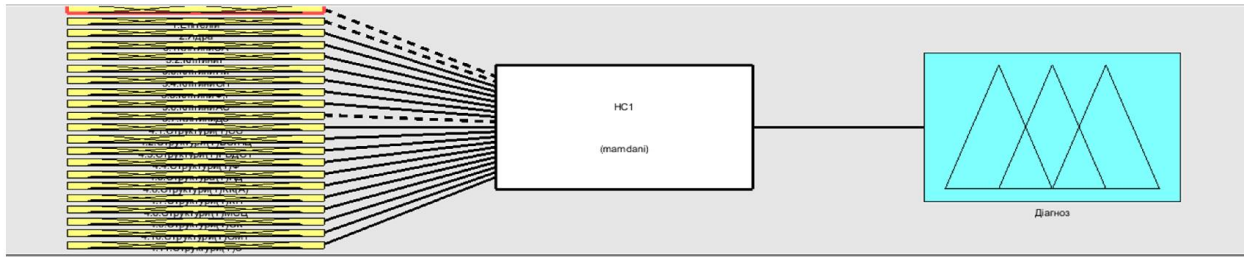


Рисунок 3.9 – Вхідні дані в редакторі «Fuzzy Logic Toolbox»

Для відображення графіків функцій належності слід вибрати необхідну змінну в лівій частині графічного інтерфейсу редактора під заголовком FIS Variables. Щоб вибрати потрібну функцію приналежності, слід натиснути на ній або її мітці в основному вікні графіків функцій належності.

Змінити вигляд функції належності можна також за допомогою маніпулятора типу миш, а саме: слід виділити змінну функцію належності на графіку (вона буде зображена червоним кольором) і, не відпускаючи натиснуту ліву кнопку маніпулятора типу миш, переміщати маркер в потрібну сторону. При цьому буде змінюватися графік відповідної функції належності і її параметри.

Вхідні дані задаються у вигляді дзвоноподібної функції належності, зображеної на рисунку 3.10, тому що накладання кількісних ознак вхідних даних одних на одні будуть мінімальними.

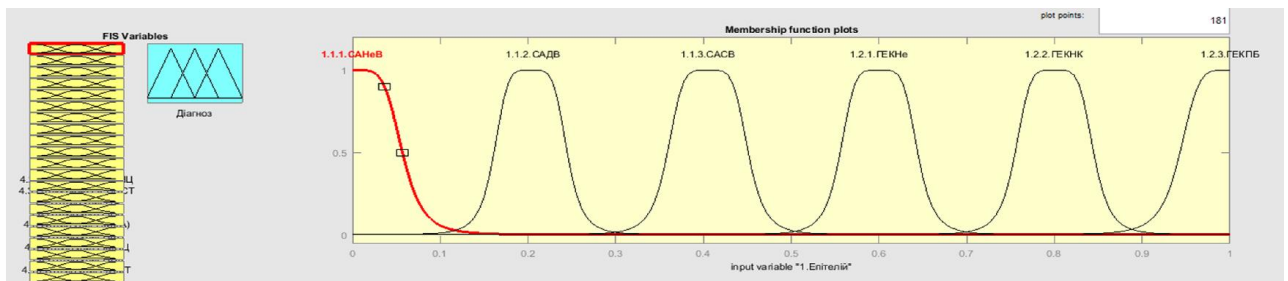


Рисунок 3.10 – Функції належності вхідних даних в редакторі «Fuzzy Logic Toolbox»

Вихідні дані виступають власне діагнозом з продукційних правил, тому вони матимуть вигляд трикутної функції належності, які зображено на рисунку 3.11.

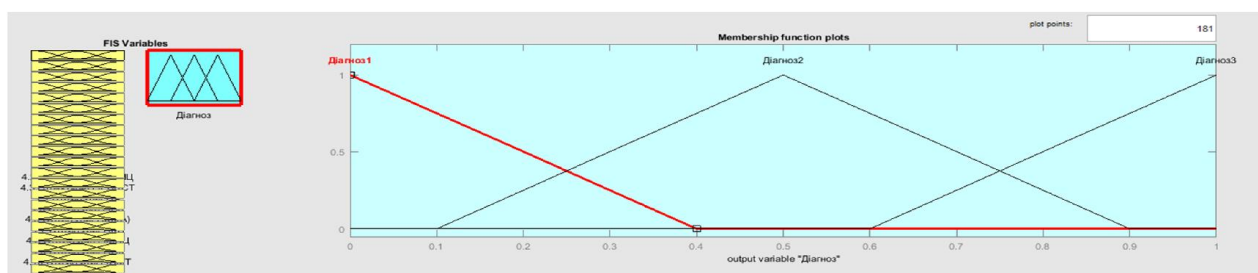


Рисунок 3.11 – Функція належності вихідних даних в редакторі «Fuzzy Logic Toolbox»

3.3 Побудова нечітких продукційних правил

Продукційні правила є невід’ємною частиною бази знань. Кожне продукційне правило відображає окрему частину знань отриманих від експерта. Продукційні правила можна модифікувати як окрему одиницю, не залежно від інших правил. Щоб створити продукційне правило необхідно відкрити редактор «Rule editor», зображено на рисунку 3.12.

Основною перевагою запису продукційних правил та формування бази знань в редакторі «Rule editor» є простота запису правил, можливість їх редагування та видалення певних правил. Крім цього запис усіх правил є однотипним.

Редактор правил системи нечіткого виведення, як випливає з його назви, призначений для завдання і редагування окремих правил системи нечіткого виведення в графічному режимі. Щоб використовувати цей редактор для створення правил, необхідно попередньо визначити всі вхідні і вихідні змінні, для чого можна скористатися редактором системи нечіткого виведення FISi редактором функцій належності. Задати правила можна за допомогою вибору відповідних значень термів вхідних і вихідних змінних.

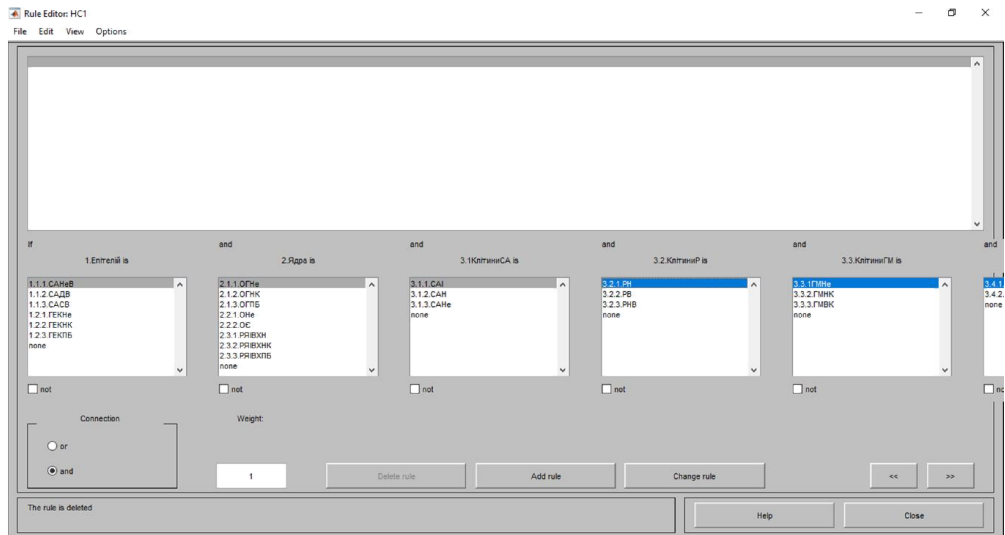


Рисунок 3.12 – Редактор «Rule editor»

Щоб створити базу продукційних правил необхідно покрокове використання кожного з вхідних терм-елементів, сукупність яких складає певну терм-множину, такий приклад наведено на рисунку 3.13.

У додатку А зображено приклад запису продукційних правил, де в центральній частині графічного інтерфейсу програми перегляду правил розташовані прямокутники, відповідні окремим вхідним змінним (функції належності жовтого кольору) і вихідним змінним (функції належності синього кольору) правил нечіткого виводу. Кожному правилу відповідає окремий рядок з цих прямокутників. Номери правил вказані в лівій частині графічного інтерфейсу.

Прямокутники вхідних змінних перетинає вертикальна пряма червоного кольору, положення якої відповідає конкретному значенню вхідної змінної відповідного стовпчика.

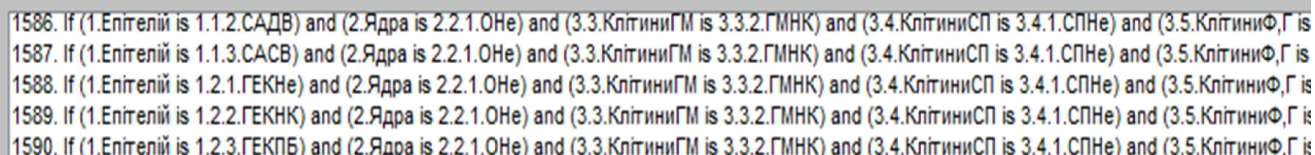
Система MATLAB реагує на кожну зміну значення окремої вхідної змінної виконанням процедури нечіткого виведення, отриманням і відображенням відповідних результуючих значень вихідних змінних

Запис продукційного правила має вигляд:

If «терм-елемент 1» and «терм-елемент 2» and ... and «терм-елемент n » than «діагноз 1».

В медичній експертній системі такі правила використовуються для встановлення зв'язків між симптомами чи якісними ознаками та діагнозом.

Для забезпечення зв'язків між елементами правила, використовуються логічні оператори «and» та «or», в даному випадку таким виступає оператор «and». Недоліком форми запису продукційних правил є те, що зв'язки між правилами і цілями, які досягаються не є явними.



```
1586. If (1.Епітелій is 1.1.2.САДВ) and (2.Ядра is 2.2.1.ОНе) and (3.3.КлітиниГМ is 3.3.2.ГМНК) and (3.4.КлітиниСП is 3.4.1.СПНе) and (3.5.КлітиниФ,Г is
1587. If (1.Епітелій is 1.1.3.САСВ) and (2.Ядра is 2.2.1.ОНе) and (3.3.КлітиниГМ is 3.3.2.ГМНК) and (3.4.КлітиниСП is 3.4.1.СПНе) and (3.5.КлітиниФ,Г is
1588. If (1.Епітелій is 1.2.1.ГЕКНе) and (2.Ядра is 2.2.1.ОНе) and (3.3.КлітиниГМ is 3.3.2.ГМНК) and (3.4.КлітиниСП is 3.4.1.СПНе) and (3.5.КлітиниФ,Г is
1589. If (1.Епітелій is 1.2.2.ГЕКНК) and (2.Ядра is 2.2.1.ОНе) and (3.3.КлітиниГМ is 3.3.2.ГМНК) and (3.4.КлітиниСП is 3.4.1.СПНе) and (3.5.КлітиниФ,Г is
1590. If (1.Епітелій is 1.2.3.ГЕКПБ) and (2.Ядра is 2.2.1.ОНе) and (3.3.КлітиниГМ is 3.3.2.ГМНК) and (3.4.КлітиниСП is 3.4.1.СПНе) and (3.5.КлітиниФ,Г is
```

Рисунок 3.13 – Приклад запису продукційних правил в редакторі «Rule editor»

Під час спільного застосування правил, вони виробляють нові знання, видаючи кращий результат, ніж результати застосування окремих правил.

Створення бази правил на основі візуальних ознак зображень передбачає багатокроковий процес, а саме: опис мікрооб'єктів та їх якісних ознак, визначення числових ознак мікрооб'єктів, визначення нечітких змінних та побудова функцій належності, формалізація правил діагностування, побудова бази нечітких правил.

База знань у продукційних системах виглядає як сукупність бази фактів і бази правил. Щоб довести правильність виконання дій згідно алгоритму побудови нечітких продукційних правил потрібно здійснити процедуру верифікації. З побудованих функцій належності впливає декілька терм-множин, які містять в собі терм-елементи.

Для побудови продукційних правил необхідно виконати усі можливі комбінації застосування кожного терм-елемента однієї терм-множини з елементами інших терм-множин.

Програма перегляду правил не дозволяє редагувати правила і функції належності термів змінних і використовується після розробки системи нечіткого

виведення на етапі її аналізу і оцінки.

В кінцевому результаті, отримані нечіткі продукційні правила встановлюють свій діагноз. Перевагою використання такого алгоритму є те, що експерт не може передбачити усіх варіантів продукційних правил, пришвидшують роботу експерта для встановлення діагнозу. Передані експерту продукційні правила ще раз переглядаються і на основі них встановлюється діагноз.

3.4 Аналіз нечітких продукційних правил

Результати роботи розробленої нечіткої системи на прикладі десяти нечітких продукційних правил наведено в таблиці 3.5.

Згенеровано Matlab код, який наведено в додатку Б.

Таблиця 3.5 – Результати нечіткої системи

№	Епітелій	Ядра	Клітини СА	Клітини Р	Клітини ГМ	Клітини СП	Клітини Г,Ф	Клітини АС	Клітини ДЗ	Структури СС	Структури ВОПЦ
1	0,045	0,046	0,062	0,063	0,139	0,063	0,009	0,172	0,134	0,092	0,136
2	0,136	0,103	0,1	0,138	0,176	0,148	0,175	0,245	0,263	0,163	0,227
3	0,227	0,231	0,336	0,219	0,250	0,245	0,090	0,3	0,318	0,120	0,336
4	0,336	0,269	0,362	0,347	0,306	0,381	0,435	0,390	0,372	0,231	0,427
5	0,391	0,383	0,4	0,451	0,565	0,390	0,527	0,627	0,467	0,361	0,518
6	0,591	0,417	0,532	0,510	0,750	0,627	0,619	0,645	0,609	0,435	0,590
7	0,736	0,657	0,792	0,685	0,787	0,7	0,694	0,736	0,627	0,731	0,681
8	0,791	0,824	0,864	0,879	0,842	0,790	0,768	0,790	0,718	0,787	0,736
9	0,918	0,917	0,913	0,864	0,880	0,912	0,842	0,845	0,891	0,865	0,863
10	0,973	0,954	0,951	0,936	0,935	0,972	0,935	0,936	0,9	0,932	0,972

Продовження таблиці 3.5

№	Структури РВДСТ	Структури Ф	Структури ПД	Структури КК(А)	Структури КП	Структури МСЦ	Структури СК	Структури ОМТ	Структури С	Структури Г	Діагноз
1	0,109	0,090	0,136	0,124	0,202	0,117	0,244	0,228	0,108	0,181	0,143
2	0,175	0,245	0,172	0,138	0,287	0,159	0,287	0,271	0,138	0,223	0,236
3	0,25	0,281	0,227	0,231	0,372	0,266	0,329	0,358	0,157	0,287	0,329
4	0,416	0,409	0,318	0,416	0,581	0,414	0,419	0,402	0,189	0,351	0,479
5	0,453	0,536	0,445	0,537	0,648	0,457	0,621	0,597	0,287	0,372	0,5
6	0,546	0,681	0,5	0,546	0,691	0,585	0,691	0,619	0,394	0,649	0,594
7	0,75	0,772	0,599	0,7	0,734	0,712	0,734	0,684	0,5	0,67	0,668
8	0,845	0,809	0,681	0,768	0,810	0,778	0,797	0,728	0,734	0,734	0,720
9	0,879	0,9	0,790	0,842	0,925	0,861	0,812	0,75	0,819	0,789	0,813
10	0,935	0,954	0,881	0,899	0,968	0,946	0,9	0,923	0,904	0,862	0,986

Точність результатів залежить від кількості вихідних даних, використуваних для побудови нечіткої системи Аналіз результатів роботи нечіткої системи підтверджує правильність її роботи. Наприклад, при сплющеному апокриновому епітелію чи гіперплазії епітеліальних клітин, що відповідає стану змінної «1.Епітелій» та дорівнює значенню 0,045 і при округлих гіперхромних ядрах та розмірі ядра з інтенсивно вираженим хроматином, що відповідає стану змінної «2.Ядра» дорівнює значенню 0,046 і при секретній активності клітини, яка відповідає стану змінної «3.1.КлітинаСА» та дорівнює 0,062 і при розмірі клітини, що відповідає стану змінної «3.2.КлітинаР» та дорівнює 0,063 і при гіперхромній мономорфності клітини, що відповідає стану змінної «3.3. КлітинаГМ» та дорівнює 0,139 і при сплющеній полігональності клітин, яка відповідає стану змінної «3.4.КлітинаСП» та дорівнює значенню 0,063 і при фагоцитах, гістоцидах, що відповідає стану змінної «3.5.КлітинаФ,Г» та дорівнює значенню 0,009 і при апокриновій секретії клітини, яка відповідає стану змінної «3.6.КлітинаАС» та дорівнює значенню 0,172 і при дистрофічних змінах клітини, що відповідає

стану змінної «3.7.КлітиниДЗ», що дорівнюють 0,134 і при сосочкових структурах, що відповідають стану змінної «4.1.Структури(Т)СС» та дорівнюють значенню 0,092 і при вузькому обідку інтенсивно пофарбованої цитоплазми, яка відповідає стану змінної «4.2.Структури(Т)ВОПЦ» та дорівнює 0,136 і при розростанні внутрішньо-долькової сполучної тканини, яка відповідає стану змінної «4.3.Структури(Т)РВДСТ» та дорівнює значенню 0,109 і при фібропластах, що відповідає стану змінної «4.4.Структури(Т)Ф» та дорівнює 0,090 і при протоках і дольках, які відповідають стану змінної «4.5.Структури(Т)ПД» і дорівнює значенню 0,136 і при клітинних комплексах (ацинусах), що відповідають стану змінної «4.6.Структури(Т)КК(А)» і дорівнюють значенню 0,124 і при клітинних пластах, які відповідають стану змінної «4.7.Структури(Т)КП» і дорівнює 0,202 і при мікросомальній структурі цитоплазми, що відповідає стану змінної «4.8.Структури(Т)МСЦ» та дорівнює значенню 0,117 і при сосочкоподібних комплексах, які відповідають стану змінної «4.9.Структури(Т)СК» та дорівнюють значенню змінної 0,244 і при однотипних мілких трубочках, які відповідають стану змінної «4.10.Структури(Т)ОМТ» та дорівнюють 0,228 і при стромі структури, що відповідає стану змінної «4.11.Структури(Т)С» і дорівнює 0,108 і при гіалінозстромі, яка відповідає стану змінній «4.11.Структури(Т)Г» та дорівнює значенню 0,181, підтверджує діагноз – рак, що відповідає стану змінної «Діагноз 1» та дорівнює значенню 0,143.

Основне призначення програми перегляду правил полягає в можливості візуалізувати результати нечіткого виведення і отримувати значення вихідних змінних в залежності від вихідних значень вхідних змінних.

Для візуалізації результати досліджень, а саме поверхню значень залежності виходу нечіткої системи від вхідних значень зображено на рисунках 3.15 – 3.16.

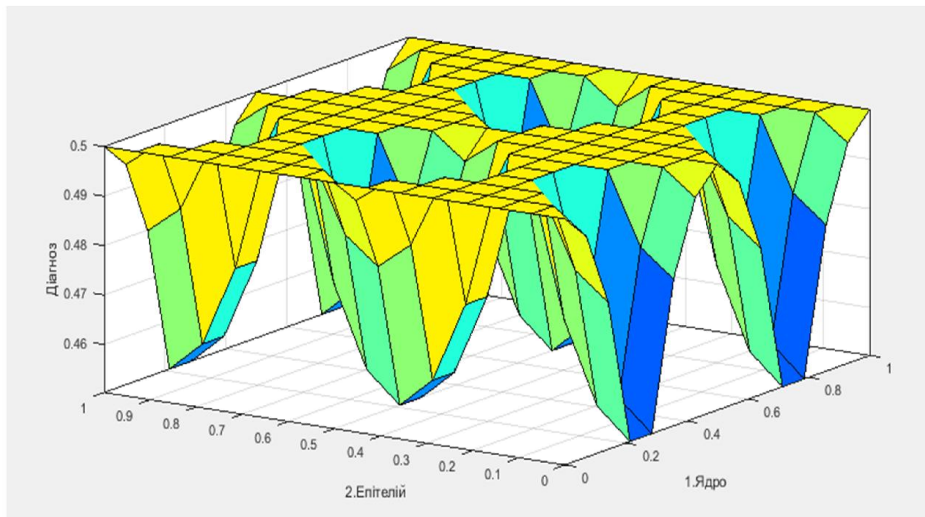


Рисунок 3.15 - Поверхня значень залежності виходу нечіткої системи від вхідних значень «Епітелій» та «Ядро»

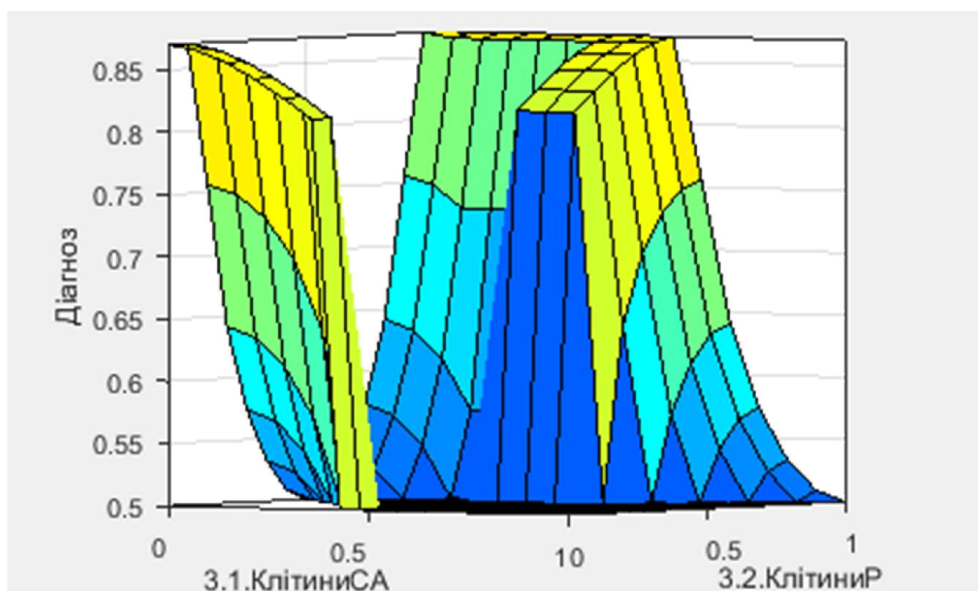


Рисунок 3.16 – Поверхня значень залежності виходу нечіткої системи від вхідних значень «Клітини секретна активність» та «Клітини розмір»

Як видно з результатів роботи нечіткої системи, визначені функції належності на основі кількісних ознак мікрооб'єктів, отриманих від експерта, побудовані функції належності в середовищі Fuzzy Logic Toolbox та нечіткі продукційні правила в середовищі Fuzzy Logic Toolbox підтверджують правильність її роботи.

ВИСНОВКИ

В результаті виконання дипломної роботи розроблено алгоритми побудови нечітких продукційних правил на основі аналізу гістологічних та цитологічних зображень, а також здійснено моделювання в середовищі Fuzzy Logic Toolbox програмного комплексу Matlab. При цьому отримано такі результати:

1) здійснено аналіз бази даних гістологічних та цитологічних зображень в результаті, отримано кількісні та якісні ознаки мікрооб'єктів;

2) проведено аналіз відомих моделей представлення знань і показано, що найбільш адекватною моделлю представлення знань є нечітка продукційна модель;

3) проведено аналіз програмних засобів представлення знань, в результаті чого вибрано Fuzzy Logic Toolbox програмного комплексу Matlab;

4) на основі нечіткої продукційної моделі розроблено узагальнений алгоритм побудови нечітких продукційних правил діагностування на основі зменшення кількості вхідних нечітких змінних;

5) на основі виділених якісних ознак побудовано функції належності лінгвістичних змінних;

6) здійснено моделювання розробленої бази нечітких продукційних правил в середовищі Fuzzy Logic Toolbox програмного комплексу Matlab;

7) здійснено аналіз отриманих нечітких продукційних правил, що дало можливість підтвердити працездатність роботи нечіткої системи діагностування ракових станів молочної залози.

Розроблена нечітка система може реалізовуватися як програмний засіб для використання в сучасних системах телемедицини.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мартинчук Т.О. Алгоритми побудови нечітких продукційних правил на основі аналізу біомедичних зображень / С.О. Вербовий, Т.О. Мартинчук // Матеріали VI Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2016. – Тернопіль: ТНЕУ, 2016. – С. 63-65.

2. Березький О.М. Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня “Магістр”. Спеціальність "Комп'ютерні системи та мережі" / О.М. Березький, Л.О. Дубчак, Г.М. Мельник / Під ред. О.М. Березького – Тернопіль: ТНЕУ, 2016 – 42 с.

3. Березький О.М. Нечітка база знань інтелектуальної системи діагностування видів раку молочної залози / О.М. Березький, Г.М. Мельник, К.М. Березька // Вісник Хмельницького національного університету - 2013. - № 6. - С. 284- 291

4. Березький О.М. База даних цитологічних та гістологічних зображень ауто- та ксеногенних тканин / Березький О.М., Мельник Г.М., Дацко Т.В., Вербовий С.О. / Науковий вісник національного лісотехнічного університету України: збірник науково-технічних праць. – Львів: РВВ НЛТУ України. – 2014.– Вип. 24.10. – С.338-345.

5. Березький О.М. Гібридна інтелектуальна інформаційна технологія опрацювання біомедичних зображень / О.М. Березький, Н.В. Білоус // Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту: Матеріали міжнародної наукової конференції. – Херсон: Видавництво ХНТУ, 2015. – С. 30-32

6. Березький О.М. Інтелектуальна система для діагностування різних форм раку молочної залози на основі аналізу гістологічних і цитологічних зображень / О.М. Березький, Г.М. Мельник, Ю.М. Батько, Т.В. Дацко // Науковий вісник НЛТУ України. – 2013 – С. 357-367.

7. Березский О.Н. Информационная технология анализа и синтеза

гистологических изображений в системах автоматизированной микроскопии / О.Н. Березский, Г.Н. Мельник // Управляющие машины и системы. – 2013. – № 3. – с.10-16

8. Абакумов В.Г. Базовые методы обработки биомедицинских изображений / В.Г Абакумов. Электроника и связь. - Киев, 2008. - № 3-4. - С. 53-56

9. Антошук С. Гибридные информационные модели в системах обработки изображений / С. Антошук, О. Бабилунга // Компьютинг – 2009. – Т. 2, № 8. – С.41-49

10. Bleau, A. Watershed-based segmentation and region merging: Computer Vision Image Understanding, A. Bleau, , L. J. Leon,- 2000.- v. 77, no. 3 - p.317–370

11. Shapiro L. G. Computer Vision/ L. G. Shapiro, G. C. Stockman - Prentice Hall, 2001 - P. 74-75

12. Gansler T. S. Breast Cancer. Atlas of Clinical Oncology/ B.C. Decker - London., 2000 - 223 с.

13. Wolberg W. H. et al "Computer-derived nuclear features distinguish malignant from benign breast cytology", Human Pathology, 26 - 1995 - P.792-796

14. Абакумов В.Г. Системы отображения в медицине / В.Г. Абакумов, А.Н. Рыбин , Й.Сватош, Ю.С Синекон - К.: Юніверс, 2001. - С. 64-69

15. Nene S. A. Columbia Object Image Library (COIL - 20). Technical Report CUCS - 005-96 / A. Nene, S. K. Nayar, H. Murase. [Электронный ресурс]. – Режим доступа: <http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php>

16. Geusebroek J. M. The Amsterdam library of object images. Int. J. Comput. Vision /J. M. Geusebroek, G. J. Burghouts, A. W. M. Smeulders. - 2005. - 61(1) - P. 103-112.

17. Jain V. The Indian Face Database. / V. Jain, A. Mukherjee [Электронный ресурс]. – Режим доступа: [http://vis-www.cs.umass.edu/~vidit/Indian Face Database](http://vis-www.cs.umass.edu/~vidit/Indian_Face_Database).

18. Bankman IN (ed). Handbook of Medical Image Processing and Analysis.

2nd ed. - New York: Academic Press,- 2008 - P.103-112

19. Jan J. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods. Boca Raton: - CRC Press; 2005 – P.38-49

20. Collection Development Manual Biomedical Imaging [Электронный ресурс]. – [Режим доступа]: <https://www.nlm.nih.gov/tsd/acquisitions/cdm/subjects15.html>

21. Заде Л. Основы нового подхода к анализу сложных систем и процессов принятия решений.— В кн.: Математика сегодня. М.: Знание, 2004 - С.5-49.

22. Петров Б. Н. Информационные аспекты качественной теории динамических систем / Б. Н. Петров, Г. М. Уланов, И. И. Гольденблат — В кн.: Техническая кибернетика. Итоги науки и техники. Т. 8. Ч. 1, 2. М.: ВИНТИ, с. 2007 – 275 с.

23. Петров Б. Н. Информационные и термодинамические аспекты качественной теории эволюционных моделей развивающихся динамических систем управления / Б. Н. Петров, Г. М. Уланов, И. И. Гольденблат.— В кн.: Техническая кибернетика. Итоги науки и техники, - т. 10. - М., ВИНТИ,. 2008 – 305 с

24. Березский О. Н. Алгоритмы анализа и синтеза биомедицинских изображений // Пробл. информатики и управления. – 2007. – № 2. – С. 134–144.

25. Путьтин Е.П., Аверин С.И. Обработка изображений в робототехнике./ Е.П. Путьтин, С.И. Аверин С.И. – М.: Машиностроение, 1990 – 320 с

26. Batié N. Entropy and fuzzy integral./ Batié N., Trillas E. - Journal of Mathematical Analysis and Applications - 2009 - v. 69 - P. 469-474

27. . Ronco A. Use of artificial neural networks in modeling associations of discriminant factors: towards an intelligent selective breast cancer screening / Alvaro L. Ronco // Artificial intelligence in medicine – № 16. – 1999. – P.299–309.

28. Bezdek J. C. Numerical taxonomy with fuzzy sets.—Journal of Mathematical Biology – 2004 - v. 1 - P. 57-71

29. Capocelli R. Fuzzy sets and decision theory. R. Capocelli, A. De Luca — Information and Control. – 2003 - v. 23 - P. 446—473
30. Chang S. K. On the execution of fuzzy programs using finite state machines.— IEEE Transactions on Computers – 2002 - v.-21 - P. 241—253
31. De Luca A. A definition of a non-probabilistic entropy in the setting of fuzzy sets theory. / A. De Luca, S. Termini— Information and Control. - 2002, v. 20 - P. 301-312
32. DeLuca A. Algebraic properties of fuzzy sets// A. De Luca, S — Journal of Mathematical Analysis and Applications – 2002 - v. 40 - P.373—386
33. De Luca A. Entropy of L-fuzzy sets / A. De Luca, S — Information and Control. – 2004 - v. 24 - P. 55—73
34. De Luca A. On the convergence of entropy measures of fuzzy sets // A. De Luca, S — Kybernetes. – 2007 - v. 6 - P. 219—227
35. De Luca A. Entropy and energy measures of a fuzzy set.— In: Advances in Fuzzy Set Theory and Applications/Ed. by M. M. Gupta, R. K. Ragade, R. R. Yager. - Amsterdam: North-Holland, 2009. – 321 p.
36. De Luca A. On some algebraic aspects of the measures of fuzziness— In: Fuzzy Information and Decision Processes/Ed. By M. M. Gupta, E. Sanchez. - Amsterdam: North-Holland, 2002. - P. 17—24
37. Di Nola A. On the fuzziness measure and negation in totally ordered lattices / A. Di Nola, S. Sessa — BUSEFAL, 2001. - v. 8 - P. 68—77
38. Di Nola A Ordering via fuzzy entropy.— In: Fuzzy Information and Decision Processes/Ed. by M. M. Gupta, E. Sanchez. Amsterdam: North-Holland, 2012, P. 25—28
39. Di Nola A. Pointwise choice criteria determined by global properties. / A. Di Nola, A. G. Ventre— BUSEFAL, 2013, v. 12 - P. 89—97
40. Di Nola A. Synthesis with respect to fuzzy entropy / A. Di Nola, A. G. Ventre — Kybernetes, v. 12, 2003. - P 205—208
41. Dumitrescu D. A definition of an informational energy in fuzzy sets

theory.— Studia University Babes — Bolyai, 2007, v. 2. - P. 57—59

42. Ebanks B. R. On measures of fuzziness and their representation.— Journal of Mathematical Analysis and Applications. - 2003, v. 94. - P. 24—37

43. Emptoz H. Nonprobabilistic entropies and indetermination measures in the setting of fuzzy sets theory.— Fuzzy Sets and Systems, 2001, v. 5. - P. 307—317.

44. Goguen J. A. L-fuzzy sets.— Journal of Mathematical Analysis and Applications, 2007, v. 18. - P.145—174

45. Higashi M. On measures of fuzziness and fuzzy complements / M. Higashi, G. J. Klir — International Journal of General Systems. 2012, v. 8. - P. 169—180

46. Higashi M. On the notion of distance representing information closeness: possibility and probability distributions / M. Higashi, G. J. Klir.— International Journal of General Systems, 2003 - v. 9. - P. 103—115

47. Ishikawa A. The fuzzy entropy concept and its application / Ishikawa A., Mieno H. — Fuzzy Sets and Systems, 2009. - v. 2. - P. 113—123

48. Штовба С.Д. Побудова функцій належності нечітких множин за кластеризацією експериментальних даних // Інформаційні технології та комп'ютерна інженерія. – 2006. – №2. – С. 92–95

49. Knopfmacher J. On measures of fuzziness.— Journal of Mathematical Analysis and Applications, 2005, v. 49. - P.529—534

50. Loo S. G. Measures of fuzziness.— Cybernetics, 2007, v. 3. - P. 201—207.

51. Pollatschek M. A. Hierarchical systems and fuzzy set theory.— Kybernetes, 2007, v. 6.- P. 147—151

52. Riera T. From measures of fuzziness to booleanity control / T. Riera, E. Triilas — In: Fuzzy Information and Decision Processes/Ed. by M. M. Gupta, E. Sanchez. Amsterdam: North-Holland, 2002, P. 3—16

53. Timothy J. R. Fuzzy logic with engineering applications / Timothy J. Ross.—3rd ed. – 2010 – 607 с.

54. Гонсалес Р. Цифровая обработка изображений в среде MATLAB. Р. Гонсалес, Р. Вудс, С. Эддинсю Москва: Техносфера, 2006. - 616 с.

55. А.Леоненков. Нечеткое моделирование в среде MATLAB и fuzzyTECH. БХВ-Петербург. 2003. –360 с.

56. Попова Г.М., Степанов В.Н. Анализ и обработка изображений медико-биологических объектов // Автоматика и телемеханика. – 2004. – №1 – С. 131-142

57. Алгоритмы нечёткого вывода: алгоритм Мамдани и алгоритм Сугэно // В. Дьяконов, В. Круглов. Математические пакеты расширения MATLAB. Специальный справочник. — Санкт-Петербург: Питер, 2001 — С. 307–309

58. Штовба С. Д. Проектирование нечетких систем средствами MATLAB / С. Штовба. — М: Горячая линия-Телеком, 2007. — 288 с.

59. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику. – Винница: Издательство винницкого государственного технического университета, 2001. – 198 с.

60. Невмержицький О.В. Аналіз сучасних моделей, орієнтованих на знання, та методів прийняття рішень / О.В. Невмержицький // Інформаційні технології проектування. – №13. – 2013. – С.119 –125