Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально науковий інститут інформаційних технологій та робототехніки

(повна назва факультету)

Кафедра комп’ютерних та інформаційних технологій і систем

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

бакалавра

(освітньо-кваліфікаційний рівень)

на тему

Розробка апаратно-програмних засобів інтелектуальної

нейромоделі контролю вартості відеоапаратури

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Виконав: студент 4 курсу, групи 402 - ТК

спеціальності

\_\_123 Комп’ютерна інженерія\_\_\_\_\_\_

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Дорошенко Є. Р.\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_Альошин С. П.\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент \_\_\_\_Бороздін М. К.\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА МЕХАНОТРОНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**спеціальність 123 «Комп’ютерна інженерія»**

**на тему**

**«Розробка апаратно-програмних засобів інтелектуальної нейромоделі контролю вартості відеоапаратури»**

**Студента групи 402-ТК Дорошенка Євгена Руслановича**

Керівник роботи

к.т.н., доцент

Альошин С.П.

Завідувач кафедри

к.т.н., доцент

Головко Г.В.

Полтава – 2021

**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 76 сторінок, 24 малюнка, 1 таблиця, 1 додаток, 27 джерел.

**Об’єкт дослідження**: інформаційне забезпечення інтелектуальної нейромоделі контролю вартості відеоапаратури.

**Предмет дослідження:** нейромережеві моделі контролю вартості відеоапаратури.

**Мета роботи**: дослідження нейромережевих технологій при діагностиці вартості відеоапаратури та розробка апаратно-програмних засобів інтелектуальної нейромоделі контролю вартості відеоапаратури, з можливістю маcштабування та додавання інших параметрів.

**Методи:** математична статистика, теорія ймовірностей, математичний аналіз, нейромережевий аналіз, класифікація, регресія, кластеризація.

**Ключові слова**: нейромережеве забезпечення, інтелектуальний класифікатор, простір ознак, нейромережеві технології,

**ЗМІСТ**

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7](#_Toc74858833)

[ВСТУП 8](#_Toc74858834)

[РОЗДІЛ 1 12](#_Toc74858835)

[ЕКСПЕРТНІ СИСТЕМИ В БАЗИСІ ІНТЕЛЕКТУЛЬНИХ СИСТЕМ 12](#_Toc74858836)

[1.1 Експертні системи 12](#_Toc74858837)

[1.2 Структура експертних систем 13](#_Toc74858838)

[1.3 Колективні експертні системи. Ситуаційний центр 16](#_Toc74858839)

[1.4 Проблеми побудови ЕС в базисі NN 21](#_Toc74858840)

[1.5 Шляхи подолання проблем в побудові ЕС базису NN 23](#_Toc74858841)

[1.6 Больцманівське навчання 27](#_Toc74858842)

[1.7 Зворотне поширення і навчання Коші 29](#_Toc74858843)

[РОЗДІЛ 2 35](#_Toc74858844)

[ЧАСОВІ РЯДИ 35](#_Toc74858845)

[2.1 Системи сучасного управління мікрокліматом 35](#_Toc74858846)

[2.2 Прості моделі прогнозування 38](#_Toc74858847)

[2.3 Середні та ковзаючи середні 39](#_Toc74858848)

[2.4 Методи Хольта та Брауна 41](#_Toc74858849)

[2.5 Метод Вінтерса 42](#_Toc74858850)

[2.6 Регресійні методи прогнозування 43](#_Toc74858851)

[РОЗДІЛ 3 46](#_Toc74858852)

[ОПИС ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ НЕЙРОМОДЕЛІ КОНТРОЛЮ ВАРТОСТІ ВІДЕОАПАРАТУРИ 46](#_Toc74858853)

[3.1 Рішення задач класифікації 47](#_Toc74858854)

[3.2 Моделювання етапів розробки ЕС для заданої предметної області. 49](#_Toc74858855)

[3.3 Постановка задачі. 50](#_Toc74858856)

[3.4 Розв’язання задачі. 51](#_Toc74858857)

[ВИСНОВКИ 61](#_Toc74858858)

[ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 63](#_Toc74858859)

[ДОДАТОК А 66](#_Toc74858860)

[Макрос аналізу вартості фотокамер 66](#_Toc74858861)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**ST Neural Networks –** нейронно-мережевої пакет фірми StatSoft, що представляє собою реалізацію всього набору нейромережевих методів аналізу даних.

**C++ –** мова програмування високого рівня з підтримкою декількох парадигм програмування.

**БП (MLP) –** багатошаровий перцептрон (Multiple Layer Perceptron).

**РБФ (RBF) –** радіальна базисна функція (Radial Basis Function).

**ІНМ (PNN)** – імовірнісна нейронна мережа (Probabilistic Neural Networks).

# ВСТУП

**Тема** даної роботи «Розробка апаратно-програмних засобів інтелектуальної нейромоделі контролю вартості відеоапаратури».

**Актуальність.** Сучасний прогрес продуктивності людини досягається тільки тоді, коли частину інтелектуального процесу беруть на себе комп'ютери. Одним із способів досягти бажаного прогресу в області, є "штучний інтелект ", коли комп'ютер обробляє не тільки однотипні, багаторазово повторювані операції, але і сам зможе навчатися. Крім того, створення повноцінного "Штучного інтелекту" відкриває перед людством нові горизонти розвитку. [1]

Одним з напрямків в області штучного інтелекту є інтелектуальні інформаційні системи. Інтелектуальні інформаційні системи - це природний наслідок розвитку нормальних інформаційних систем. Вони сформували в собі найбільш наукомісткі технології з найвищім ступенем автоматизації не тільки процесів підготовки інформації для прийняття рішень, але і сам хід вироблення варіантів рішень, спираючись на отримані інформаційною системою дані. [2]

Штучний інтелект з'явився на базі обчислювальної техніки, математичної логіки, програмування, психології, лінгвістики, нейрофізіології та інших галузей знань. Штучний інтелект - це зразок міждисциплінарних досліджень, де з'єднуються професійні інтереси фахівців різного профілю. сама назва нової науки виникло в кінці 60-х рр., а в 1969 року у Вашингтоні (США) відбулася перша Всесвітня конференція з питань штучного інтелекту.

Якщо дотримані дві необхідні умови, все наукове дослідження може отримати наукові права: в цих дослідженнях повинен бути об'єкт дослідження, відмінний від об'єктів, що вивчаються іншими науками; повинен бути особливий метод вивчення цього об'єкта, який відрізняється від других. Научний метод був встановлений і тепер дослідження об'єднані з терміном «штучний інтелект», з його власними конкретними об'єктами дослідження і конкретними методами.

Можна сформулювати основні цілі і завдання штучного інтелекту. Об'єктом дослідження штучного інтелекту є метапроцедури, який використовується для вирішення людських проблем, який традиційно називають інтелектом або творчістю. Але якщо психологія мислення вивчає ці метапроцедури, що відносяться до людей, то штучний інтелект створить програмні та апаратно-програмні моделі цих метапроцедур.

Метою досліджень в області штучного інтелекту є створення бібліотеки метапрограмм, достатньою для того, щоб технічні системи могли знаходити рішення на основі постановок задач. Поставлені цілі не вичерпують всіх завдань, поставлених штучним інтелектом. Це найостанніша мета. Наступні цілі пов'язані зі спробами проникнути в сферу людського мислення, яка виходить за рамки раціонально-виразною області мовного (лінгвістичного) мислення. При знаходити власні шляхи розв'язання багатьох проблем, особливо відмінних від раніше вирішених, область мислення, також відома як підсвідомість, несвідоме або інтуїція, також відіграє важливу роль.

Основними методами, використовуваними в штучному інтелекті, є різні програмні моделі і інструменти, комп'ютерні експерименти і теоретичні моделі. Однак комп'ютери досягли своєї межі в створенні штучного інтелекту. Ми шукаємо нові технічні структури, які можуть краще вирішувати завдання, пов'язані з інтелектуальним процесом. Сюди входить вивчення нейронних мереж.

У дослідженнях штучного інтелекту є кілька основних проблем:

1. Подання знань - розробка методів і прийомів формалізації, а потім введення систем знань з різних проблемних областей в пам'ять, накопичення узагальнень і класифікації знань, а також застосування знань при вирішенні проблем.
2. Моделювання міркувань - дослідження і формалізація різних схем міркувань людини, що використовуються в процесі вирішення різних завдань, для створення ефективних програм реалізації цих схем на комп'ютері.
3. Діалогова програма спілкування природною мовою забезпечує зв'язок між інтелектуальними системами і людьми-експертами в процесі вирішення проблем.
4. Планування доцільної діяльності - розробка методів побудови програм складної діяльності на підставі тих знань про проблемну область, які зберігаються в інтелектуальній системі.
5. Навчання інтелектуальних систем в процесі їх діяльності, створення комплексу засобів для накопичення і узагальнення умінь і навичок, які накопичуються в таких системах.

Крім цих проблем досліджуються багато інших, становлять той заділ, на який будуть спиратися фахівці на наступному витку розвитку теорії штучного інтелекту. [3]

**Метою** даної роботи є дослідження нейромережевих технологій при діагностиці вартості відеоапаратури та розробка апаратно-програмних засобів інтелектуальної нейромоделі контролю вартості відеоапаратури, з можливістю маcштабування та додавання інших параметрів.

Для досягнення мети необхідно вирішити такі завдання:

* дослідити недоліки існуючих нейромережевих технологій;
* знайти підходи до виправлення недоліків існуючих нейро-мережевих технологій, не втрачаючи при цьому продуктивність;
* проаналізувати різноманіття ознак при діагностиці вартості відеоапаратури та проблеми оцінки їх інформативності;
* розробити на основі вивченого матеріалу алгоритм для реалізації технології;
* сформувати набір навчальних даних;
* створити ансамбль нейромережевих технологій, що задовольняють задачу визначення стану самопочуття організму;
* провести аналіз отриманих результатів.

**Об'єктом** роботи є інтелектуальна нейромодель.

**Предметом** роботи є інтелектуальна нейромодель контролю вартості відеоапаратури.

У **вступі** формується тема, актуальність, основні проблеми штучного інтелекту, мета, завдання для досягнення мети, а також об'єкт і предмет дипломної роботи.

**У висновку** формуються загальні висновки по роботі.

# РОЗДІЛ 1

# ЕКСПЕРТНІ СИСТЕМИ В БАЗИСІ ІНТЕЛЕКТУЛЬНИХ СИСТЕМ

## **1.1 Експертні системи**

Експертна система (ЕС) - програмно-технічний інструмент, що дозволяє користувачеві в діалоговому режимі одержувати від комп'ютера консультаційну допомогу в конкретній предметній області, де сконцентровані навики і знання людей-експертів (фахівців в даній області). Мета досліджень по ЕС полягає в розробці програм (пристроїв), які при вирішенні завдань, важких для експерта-людини, отримують результати, що не поступаються за якістю і ефективності рішень, що отримуються експертом. У більшості випадків експертні системи вирішують важкі доручення або задачі, що не мають алгоритмічного рішення.

В основі роботи ЕС лежить застосування знань, а маніпулювання ними здійснюється на базі евристичних правил, сформульованих експертами. ЕС надають поради, дають консультації, виконують класифікацію, проводять аналіз і ставлять діагноз. Вони орієнтовані на рішення задач, що зазвичай вимагають проведення експертизи людиною-фахівцем. На відміну від машинних програм, що використовує процедурний аналіз, ЕС вирішують задачу у вузькій предметній області (конкретної галузі експертизи) на основі дедуктивних міркувань. Головний плюс експертних систем - змога нагромаджувати досвід, зберігати їх довгий період, оновляти і тим самим забезпечувати відносну самостійність конкретної організації від наявності в ній кваліфікованих фахівців .

Області застосування ЕС , заснованих на знаннях, можуть бути згруповані в кілька основних класів: медична діагностика, контроль та управління, діагностика несправностей в механічних і електричних пристроях, навчання.

1. ***Медична діагностика***. Діагностичні системи використовуються для встановлення зв'язку між порушеннями діяльності організму і їх можливими причинами.

2. ***Прогнозування***. Прогнозуючі системи передбачають можливі результат події на основі даних про поточний стан об'єкта. Навіть на персональному комп'ютері, встановивши просту систему, засновану на знаннях, ви можете одержувати місцевий прогноз погоди.

3. ***Планування***. Плануючі системи призначені для звершення конкретних цілей при вирішенні завдань з великим числом змінних.

4. ***Інтерпретація***. Інтерпретуючі системи мають вміння отримувати певні висновки на основі результатів спостереження. Для прикладу визначає місцеположення і типи суден в океані за даними акустичних систем стеження.

5. ***Контроль і управління***. Контроль і управління. Системи, засновані на знаннях, можуть використовуватися в якості інтелектуальних систем контролю і затверджувати рішення, аналізуючи дані, що надходять від декількох джерел. Такі системи вже працюють на атомних електростанціях, управляють повітряним рухом і здійснюють медичний контроль. Діагностика несправностей в механічних і електричних пристроях. [4]

## **1.2 Структура експертних систем**

Узагальнена будова експертної системи представлена на Рисунку 1.1. Слід врахувати, що реальні ЕС можуть мати складнішу структуру, проте блоки, зображені на малюнку, обов’язково присутні в будь-який істинно експертної системі, тому що являють собою шаблон структури сучасної ЕС.

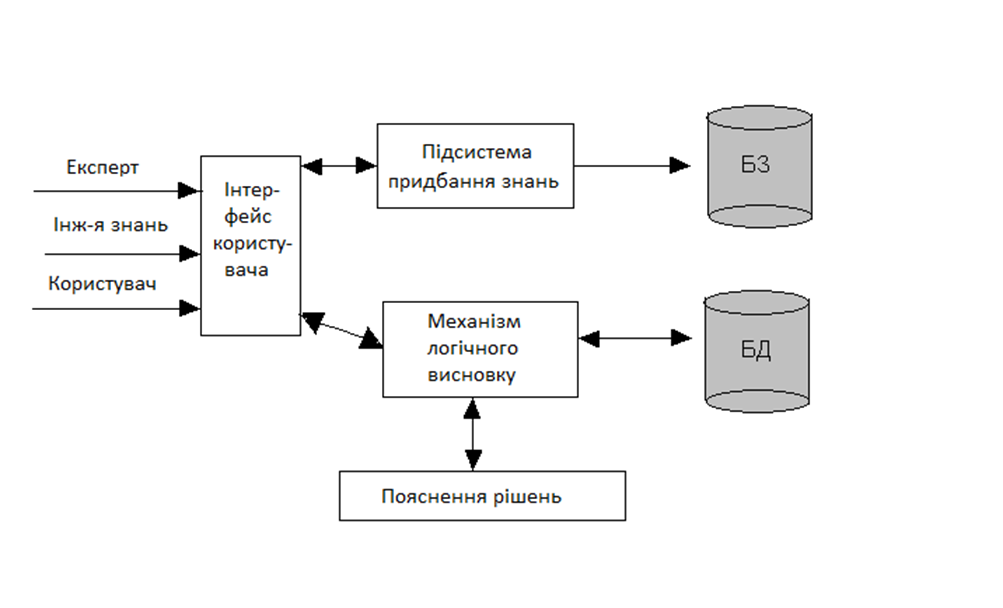


Рисунок 1.1 - Узагальнена структура експертної системи

Експертні системи мають дві категорії користувачів і два окремих "входу", що відповідають різним цілям взаємодії користувачів з ЕС:

1. ***Експерт***, якому потрібна консультація ЕС — діалоговий сеанс роботи з нею, в процесі якої вона розв'язувати експертну задачу. Діалог з ЕС здійснюється через діалоговий процесор — спеціальну компоненту ЕС. Існують дві основні форми діалогу з ЕС — діалог на обмеженій підмножині природної мови (з використанням словника-меню (при якому на кожному кроці діалогу система пропонує альтернативу професійного лексикону експертів) і діалог на основі з декількох можливих дій);
2. ***Експертна група*** інженерії знань, що складається з експертів в предметній області й інженерів знань. У функції цієї групи входить заповнення бази знань, здійснюване за допомогою спеціалізованої діалогової компоненти ЕС — підсистеми здобуття знань, яка дозволяє більш-менш автоматизувати цей процес.
3. ***Інтерфейс користувача***. Інтерфейс користувача — це система програмних і апаратних засобів, що забезпечують для кінцевого користувача застосування комп'ютера для вирішення завдань, які виникають в середовищі його професійної діяльності або без посередників, або з незначною їх допомогою. Це сукупність засобів інтелектуального інтерфейсу, що мають гнучку структуру, яка забезпечує змогу адаптації в широкому спектрі інтересів кінцевих користувачів.
4. ***Підсистема здобуття знань.*** Підсистема здобуття знань призначена для додавання в базу знань нових правил і модифікації наявних. В її розпорядження входить приведення правила до виду, що дозволяє підсистемі висновку застосовувати це правило в процесі роботи. У більш складних системах передбачені ще й кошти для перевірки введених або модифікуються правил на несуперечність з наявними правилами.
5. ***База знань.*** База знань — важлива компонента експертної системи, вона призначена для зберігання довгострокових даних, що описують розглянуту предметну сферу (а не поточних даних), і правил, що описують доцільні перебудови даних цієї області. На відміну від всіх інших компонент ЕС, база знань - "змінна" частина системи, яка може поповнюватися і модифікуватися інженерами знань і досвіду користування ЕС, між консультаціями (а в деяких системах і в процесі консультації). Існує декілька способів подачі знань в ЕС, однак загальним для всіх них є те, що знання представлені в символьній формі (елементарними компонентами представлення знань є тексти, списки та інші символьні структури). Тим самим, в ЕС реалізується принцип символьної природи міркувань, який полягає в тому, що процес міркування представляється як систематичність символьних перетворень. Існують динамічні й статичні бази знань. Динамічна база знань змінюється з часом. Її вміст залежить і від стану навколишнього. Нові факти, що додаються в базу знань, є результатом виведення, який складається в застосуванні правил до наявних фактів. У системах з монотонним висновком факти, збережені в базі знань, статичні, тобто не змінюються в процесі виконання завдання. У системах з немонотонним висновком допускається модифікація або видалення фактів з бази знань. Як приклад системи з немонотонним висновком дозволено привести ЕС, призначену для складання перспективного плану капіталовкладення компанії. У такій системі за вашим бажанням можуть бути змінені навіть ті показники, які після виведення вже викликали спрацьовування будь-яких правил. Іншими словами є можливість модифікувати значення атрибутів в складі фактів, які перебувають в робочій пам'яті. Анулювання фактів в свою чергу призводить до необхідності видалення з бази знань висновків, отриманих за допомогою згаданих правил. Тим самим рішення виконується ще раз для того, щоб переглянути ті рішення, які були отримані на основі зазнали зміни фактів.
6. ***База даних.*** База даних (робоча пам'ять) призначена для зберігання вихідних і проміжних даних розв'язуваної в поточний момент завдання.
7. ***Механізм логічного висновку.*** Основу ЕС складає підсистема логічного висновку, яка використовує інформацію з бази знань (БЗ), генерує рекомендації щодо вирішення шуканого завдання. Переважно для представлення знань в ЕС використовуються системи продукції і семантичні мережі. Припустимо, БЗ складається з фактів і правил (якщо <посилка>; то <висновок>;). Якщо ЕС визначає, що посилка вірна, то правило визнається відповідним для даної консультації та воно запускається в дію. Запуск правила означає затвердження укладення даного правила в якості складової частини процесу консультації. [5]

## **1.3 Колективні експертні системи. Ситуаційний центр**

**Ситуаційний центр (СЦ) -** це організаційна структура, яка виконує комплекс дій з аналізу ситуацій і затвердження рішень щодо ефективного управління різними процесами, що становить собою інтегровану сукупність експертів та інформаційних систем. Об'єктами управління СЦ можуть бути галузі, корпорації, організації, підприємства в широкому спектрі предметних областей від систем озброєння до систем освіти та охорони здоров'я. Характерними особливостями подібних об'єктів є важко формалізуються системоутворюючі процеси, які, як правило, не піддаються аналітичного подання, що перешкоджає оперативному виявленню і формалізації основних чинників, зв'язків між ними й сили впливу окремих факторів на кінцевий результат. Ситуація ускладнюється тим, що в ряді випадків СЦ, через особливості контрольованої предметної області, має реагувати в режимі реального часу. Це характерно, наприклад, для керування системою енергопостачання регіону, системами управління зброєю на театрі військових дій, для систем забезпечення безпеки польотів цивільної авіації, для У найзагальнішому вигляді ситуаційним центром (кімнатою або залом) можна назвати приміщення, де спостерігається поточна або аналізується можлива ситуація. Однак при такому підході будь-яка кімната, в якій знаходиться спостерігач і телевізор, передає новини про ситуацію в країні, буде вважатися ситуаційним залом. Якщо ж там є ще і радіо, телефон, факс, комп'ютер і географічна карта, то це буде персональний СЦ.

Якщо поставити акценти тільки на розгляді питань технічного оснащення, то можна сформулювати «технічне» визначення СЦ: приміщення, оснащене потужною і сучасною презентаційною технікою.

Ще однією крайністю є «програмне» визначення: СЦ — програмна і технічне середовище, яка дозволяє проводити аналіз за допомогою безлічі різних програм і інформаційних технологій. Такі системи краще називати центрами підтримки прийняття рішення або аналітичними центрами. ЦСО — інформаційна, програмна та інструментальне середовище, яка дозволяє в реальному часі перетворювати масив вхідних даних в значення прогнозованих змінних (індикаторів) і по їх сукупності розпізнавати майбутнє стан досліджуваного об'єкта або процесу на різну глибину прогнозу.

Всі три зазначених вище визначення мають одну загальну особливість — вони не розглядають питання про те, хто або що формує ситуації й хто здійснює їх аналіз. У цьому аспекті СЦ можна розділити на зовнішні та внутрішні. зовнішні СЦ служать технічної або інформаційним середовищем, необхідної оперативному персоналу для оцінки ситуації. внутрішні СЦ оперують поняттям ситуації на рівні відбиття, моделювання, аналізу чи управління. Фактично внутрішні СЦ автоматизують обробку самої ситуації, а зовнішні — вихідних даних, необхідних для її виявлення та аналізу. Можна дати такі визначення СЦ (внутрішнього):

* СЦ – сукупність програмно-технічних засобів, науково-математичних чеських методів і інженерних рішень для автоматизації процесів відбиття, моделювання, аналізу ситуацій і управління.
* СЦ — сукупність різних ССМ, науково-математичних методів і інженерних рішень для автоматизації процесів управління.

Структура СЦ, як і будь-який АСУ, включає різні види забезпечення (програмне, технічне, лінгвістичне та т. При цьому вони розглядаються на чотирьох основних рівнях: науково-математичному, інженерному, програмному і технічному. Науково-математичний рівень являє собою сукупність наукових теорій, методів, алгоритмів, досліджень і розробок, необхідних для реалізації інших рівнів. Він дозволяє обґрунтувати доцільність створення СЦ, визначити ефективність його функціонування, інтегрувати різнорідні компоненти, своєчасно виправити помилки.

Інженерний рівень охоплює конкретні апаратно-програмні засоби, а також необхідні технологічні та конструкторські розрахунки, моделі технічних пристроїв і приміщень, специфікації програм, алгоритми роботи й т.

Програмний і технічний рівні містять відповідне забезпечення, необхідне для реалізації поставлених на верхніх рівнях завдань і функцій. Згадані вище рівні охоплюють такі обов'язкові компоненти:

* + вимірювальний (сенсорна середа);
  + інформаційний (ситуаційна або імітаційна модель середовища);
  + середовище інформаційної підтримки;
  + середа апаратної підтримки;
  + середовища візуалізації;
  + оперативний склад.

Під вимірювальним (або сенсорним) середовищем СЦ розуміється сукупність апаратно-програмних засобів, службовців для отримання інформації про стан контрольованого середовища. Це можуть бути антенні системи, канали зв'язку, відео- і аудіопередач, датчики і т.

Інформаційна (ситуаційна або імітаційна) модель середовища являє собою сукупність модельованих понять проблемного середовища, просторові відносини між об'єктами моделі та відображення об'єктів моделі в безліч графічних примітивів.

Середовище інформаційної підтримки — це сукупність програм і інформаційних потоків, що забезпечують функціонування інформаційної моделі та середовища візуалізації СЦ. В першу чергу сюди входять ССМ, експертні системи й системи імітаційного моделювання. Характерною рисою будь-якого СЦ є прив'язка ситуаційної моделі до місцевості, тому в його склад можуть входити геоінформаційні системи. Для оцінки розвитку ситуацій застосовуються, зокрема, системи прогнозування на базі нейронних мереж і генетичних алгоритмів. Ефективність графічного і текстового подання може досягатися завдяки використання фрактальної та когнітивної графіки.



Рисунок 1.2 – Приклад планування приміщень ситуаціного центру:

1 – ситуаційний хал; 2 – операторська; 3 – зал аналітиків; 4 – технічне приміщення для розміщення обладнання; 5 – технічна зона для обслуговування відеостіни

**Класифікація СЦ**

Ситуаційні центри можна класифікувати по ряду ознак.

1. По складу ССМ:
   * СЦ спостереження (відображення);
   * аналітичні СЦ;
   * повнофункціональні СЦ. Поєднують функції відображення, моделювання та аналізу ситуацій.
2. За масштабом:
   * стратегічні СЦ;
   * оперативні СЦ;
   * персональні СЦ.

Стратегічний ситуаційний центр вирішує складні, масштабні, відповідальні завдання, спрямовані на структурну і функціональну перебудову. СЦ цього типу налаштовані на такі об'єкти, як галузь, регіон, велике підприємство (холдинг), відомство, складний, розподілений в просторі процес. Одним із прикладів таких СЦ є центр стратегічного моделювання IBS\_CSM.

Оперативний СЦ вирішує завдання автоматичної передачі оперативної інформації в ситуаційну модель, що дає першій особі можливість оперувати «модулями» свого бізнесу в реальному часі. Об'єктами таких СЦ є підприємство, завдання, процес, кампанія, проєкт, велика акція, однорідна функція значних масштабів (наприклад, забезпечення працездатності ретрансляційного обладнання магістралі зв'язку).

Персональний ситуаційний центр вирішує завдання експрес-оцінки ситуації, оперативного доступу до керованого об'єкта і дозволяє першому керівнику завжди «бути в курсі подій» незалежно від часу, місця (і навіть у відомому сенсі стану) керуючого суб'єкта. Персональні СЦ індиферентні стосовно масштабів керованого об'єкта; їх завдання, функції та склад визначаються скоріше суб'єктом, вирішальним, яка інформація йому знадобиться.

В силу жорстких часових обмежень на прийняття рішень, а також з-за високої динаміки даних про об'єкт і середовищі, в якій він функціонує, побудова точної моделі поведінки такого об'єкта і забезпечення ефективного управління ним залишається сьогодні актуальною проблемою. Однак, у зазначених об'єктах управління, як правило, є значний масив апріорних і поточних даних, що є об'єктивною основою для продуктивного застосування інтелектуальних технологій для трансформації бази даних в базу знань. Тоді аналітичний підхід, заснований на прецедентах, що містять інформацію про закономірності факторів і станів, їх взаємний вплив, силу та напрямок впливу, для прийняття рішень, стає винятково важливим. Таким чином, проблема прийняття рішення в СЦ зводиться до реалізації зазначених функцій шляхом створення алгоритмів і програм відображення масиву вхідних даних на стану і динаміку їх еволюції під впливом керуючих факторів. Якщо це завдання має рішення, то ситуаційний центр, оперативно (час прийняття рішення визначається технічними можливостями обчислювачів) ідентифікує стан об'єкта, прогнозує його динаміку і визначає керуючі фактори, адекватні цільовим станом. [6]

## **1.4 Проблеми побудови ЕС в базисі NN**

***Труднощі побудови ЕС***

Не зважаючи на численні успішні застосування зворотного поширення, воно не є панацеєю. Найбільше неприємностей приносить не визначено тривалий навчальний процес. У ускладнених навчальних завданнях мережі будуть корисними дні або навіть тижні, але нейромережа може і взагалі не навчитися. Тривалий час навчання може бути результатом неоптимального вибору довжини кроку. Невдачі в навчанні зазвичай виникають з двох причин: паралічу мережі та попадання в локальний мінімум.

***Параліч мережі***

Значення ваг в результаті корекції може суттєво зрости у процесі навчання. У наслідок цього всі або більшість нейронів будуть функціонувати при дуже великих значеннях OUT, в області, де похідна стискає функції дуже мала. Через це, що поверення їх у процесі навчання єпомилкою пропорційною цієї похідної, то процес навчання буде наближеним до зупунки. Теоретично, ця проблема вивчена не досконало. Тому цю проблему вирішують зменшенням розміру кроку, але це спричиняє збільшення часу навчання. Різні євристики застосовувалися для уникнення паралічу, а також для відновлення після нього. Сьогодні вони розглядаються як експериментальні.

***Локальні мінімуми***

Зворотнє поширення здійснює спуск поверхнею помилки, постійно підлаштовуючи ваги до мінімуму, що є різновидом градієнтного спуску. . Площина помилки складної мережі суттєво порізана і складається з пагорбів, складок, ярів і долин в просторі високої розмірності. Існує можливість потрапляння мережі в локальний мінімум, якщо поблизу є значно глибший мінімум. Всі напрямки спрямовані вгору, якщоточка знаходиться в локальному мінімумі, і мережа не здатна з нього вибратися. Статистичні методи навчання можуть допомогти уникнути цієї пастки, але вони повільні. Існує метод, який об'єднує статистичні методи машини Коші з градієнтним спуском зворотного поширення і приводить до системи, яка знаходить глобальний мінімум, зберігаючи високу швидкість зворотного поширення.

***Розмір кроку***

Детальний розбір доказів збіжності показує, що корекції ваг передбачаються нескінченно малими. Зрозуміло, що це практично не можливо, бо час навчання буде нескінченним. Розмір кроку потрібно обирати кінцевий, і в цьому питанні доводиться спиратися тільки на досвід. У випадку малого кроку збіжність занадто повільна, у разі великого — ймовірність виникнення паралічу, а також постійної нестійкості. Існують адаптивні алгоритми вибору кроку, автоматично коригувальні розміри кроку в процесі навчання.

***Тимчасова нестійкість***

Під час вивчення мережею розпізнавання букв недоцільно вчити «А», коли забувається «Б».Навчальний процес має бути побудований на навчані мережі на всій навчальній множині без пропусків вивченого. Ця умова виконується у доведенні збіжності , за умови надання мережі всіх векторів навчальної множини до моменту виконання корекції ваг. Необхідні зміни ваг повинні обчислюватися на всій множині, а це вимагає додаткової пам'яті; після ряду таких навчальних циклів ваги зійдуться до мінімальної помилку. Цей метод може виявитися марним, якщо мережа знаходиться в постійно мінливого зовнішнього середовища, так що другий раз один і той же вектор може вже не повториться. В цьому випадку процес навчання може ніколи не зійтися, безцільно блукаючи або сильно осцілліруя.

***Стохастичні методи***

Як для навчання штучних нейронних мереж будуть корисні стохастичні методи, так само як і для отримання виходу попередньонавченої мережі. Ці методи навчання будуть надзвичайно корисними для виключення локальних мінімумів під час навчання. Але з ними також пов'язаний ряд проблем. [7]

## **1.5 Шляхи подолання проблем в побудові ЕС базису NN**

За допомогою певного процесу штучна нейронна мережа навчається модифікувати її ваги. У разі успішного навчання, надання мережі безлічі вхідних сигналів з'являться бажані множини вихідних сигналів. Існує два класи навчальних методів: **стохастичний і детерміністський**.

**Стохастичні** методи навчання виконують псевдовипадкові зміни величин ваг, зберігаючи ті зміни, які ведуть до поліпшень. Щоб побачити, як це може бути зроблено, розглянемо рис. 1, на якому зображена типова мережа, в якій нейрони з'єднані за допомогою ваг. Вихід нейрона є тут зваженою сумою його входів, яка, перетворена за допомогою нелінійної функції. Для навчання мережі може бути використана наступна процедура:

* + Вибрати ваги випадковим чином і скорегувати їх на невелике випадкове значення. Пред'явити множину входів і обчислити виходять виходи.
  + Порівняти ці виходи з бажаними виходами і обчислити величину різниці між ними. Загальноприйнятий метод полягає в знаходженні різниці між фактичним і бажаним виходами для кожного елемента навченою пари, зведення різниць в квадрат і знаходження суми цих квадратів. Метою навчання є мінімізація цієї різниці, часто **званої цільової функцією**.
  + Вибрати ваги випадковим чином і скорегувати їх на невелике випадкове значення. Якщо корекція допомагає (зменшує цільову функцію), то зберегти її, в іншому випадку повернутися до первісного значення ваги.
  + Повторювати кроки з 1 до 3 до тих пір, поки мережа не буде навчена в достатній мірі.

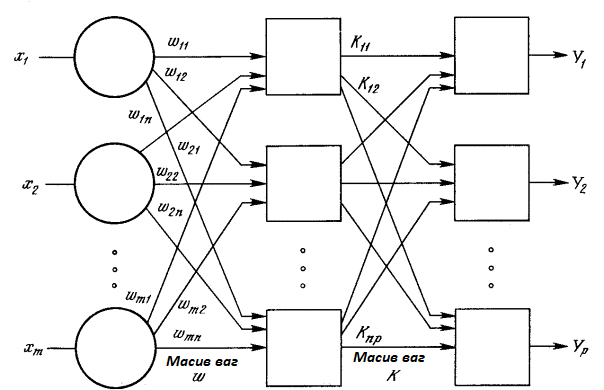


Рисунок 1.3 - Двухслойная мережа без зворотних зв'язків

**Детерміністський** метод навчання поступово здійснює процес корекції ваги мережі, основою якої є використання їх поточних значень, а також вхідних величин, фактичних виходів і бажаних виходів. Навчання персептрона є прикладом подібного підходу детермініста.

Цей процес прагне мінімізувати цільову функцію, але може потрапити, як в пастку, в невдале рішення. На Рисунку 1.4 показано, як це може мати місце в системі з єдиним вагою. Припустимо, що спочатку вага взята рівним значенню в точці А. Якщо випадкові кроки по вазі малі, то будь-які відхилення від точки А збільшують цільову функцію і будуть відкинуті. Краще значення ваги, що приймається в точці В, ніколи не буде знайдено, і система буде спіймана в пастку локальним мінімумом, замість глобального мінімуму в точці В. Якщо ж випадкові корекції ваги дуже великі, то як точка А, так і точка В будуть часто відвідувати , але те ж саме буде мати місце і для кожної іншої точки. Вага буде змінюватися так різко, що він ніколи не встановиться в бажаному мінімумі.

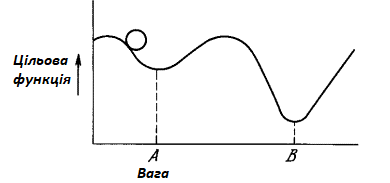


Рисунок 1.4 - Проблема локальних мінімумів.

Корисна стратегія для уникнення подібних проблем полягає в великих початкових кроках і поступовому зменшенні розміру середнього випадкового кроку. Це дозволяє мережі вириватися з локальних мінімумів і в той же час гарантує остаточну стабілізацію мережі.

Пастки локальних мінімумів докучають інстанціям алгоритмам навчання, заснованим на пошуку мінімуму, включаючи персептрон і мережі зворотного поширення, і представляють серйозну і широко поширену трудність, якої часто не помічають. Стохастичні методи дозволяють вирішити цю проблему. Стратегія корекції ваг, яка змушує ваги приймати значення глобального оптимуму в точці В, можлива.

Як пояснює аналогією припустимо, що на рис. 2 зображений кульку на поверхні в коробці. Якщо коробку сильно потрясти в горизонтальному напрямку, то кулька буде швидко перекочуватися від одного краю до іншого. Ніде не затримуючись, в кожний момент кулька буде з рівною імовірністю знаходитися в будь-якій точці поверхні.

Якщо поступово зменшувати силу струшування, то буде досягнута умова, при якому кулька буде на короткий час «застрявати» в точці В. При ще більш слабкому струшуванні кулька буде на короткий час зупинятися як в точці А, так і в точці В. При безперервному зменшенні сили струшування буде досягнута критична точка, коли сила струшування достатня для переміщення кульки з точки А в точку В, але недостатня для того, щоб кулька могла видертися з В в А. Таким чином, остаточно кулька зупиниться в точці глобального мінімуму, коли амплітуда струшування поменшає до нуля.

Штучні нейронні мережі можуть навчатися по суті тим же самим чином за допомогою випадкової корекції ваг. Спочатку робляться великі випадкові корекції із збереженням тільки тих змін ваг, які зменшують цільову функцію. Потім середній розмір кроку поступово зменшується, і глобальний мінімум в кінці кінців досягається.

Це сильно нагадує відпал металу, тому для її опису часто використовують термін «імітація відпалу». У металі, нагрітому до температури, що перевищує його точку плавлення, атоми знаходяться в сильному безладному русі. Як і у всіх фізичних системах, атоми прагнуть до стану мінімуму енергії (єдиного кристалу в даному випадку), але при високих температурах енергія атомних рухів перешкоджає цьому. У процесі поступового охолодження металу виникають все більш низькоенергетичні стану, поки, зрештою, не буде досягнуто найнижча з можливих станів, глобальний мінімум. У процесі відпалу розподіл енергетичних рівнів описується наступним співвідношенням:

                                               P(e) = exp(–e/kT), (1)

де Р(е) - імовірність того, що система знаходиться в стані з енергією е; k - постійна Больцмана; Т - температура за шкалою Кельвіна.

При високих температурах Р(е) наближається до одиниці для всіх енергетичних станів. Таким чином, високоенергетичне стан майже настільки ж ймовірно, як і низькоенергетичний. У міру зменшення температури ймовірність високоенергетичних станів зменшується в порівнянні з низькоенергетичними. При наближенні температури до нуля стає дуже малоймовірним, щоб система перебувала в високоенергетичному стані. [8]

## **1.6 Больцманівське навчання**

Так як потрібна величина зміни ваги w , а не ймовірність зміни ваги, що має величину w, то метод Монте-Карло може бути використаний в такий спосіб:

1. Знайти кумулятивну ймовірність, відповідну P(w). Це є інтеграл від P(w) в межах від 0 до w. Так як в даному випадку P(w) не може бути проінтегрована аналітично, вона повинна інтегруватися чисельно, а результат необхідно затабуліровать.
2. Вибрати випадкове число з рівномірного розподілу на інтервалі (0,1). Використовуючи цю величину як значення P(w), знайти в таблиці відповідне значення для величини зміни ваги.

Властивості машини Больцмана широко вивчалися. Швидкість зменшення температури повинна бути обернено пропорційна логарифму часу, щоб була досягнута збіжність до глобального мінімуму. Швидкість охолодження в такій системі виражається таким чином:

(2)

де - штучна температура як функція часу; - початкова штучна температура; t - штучний час.

Цей розчаровуючий результат пророкує дуже повільну швидкість охолодження (і дані обчислення). Цей висновок підтвердився експериментально. Машини Больцмана часто вимагають для навчання дуже великого ресурсу часу.

У багатьох практичних ситуаціях функціональний зв'язок між незалежними змінними і цільовою функцією невідома і фактично не може бути відомою. Складний хімічний процес може не мати адекватної математичної моделі. Єдиними вимірюваними величинами можуть бути «вихід», «якість», «ціна» і т.д., Які є невідомими функціями від великого числа таких незалежних змінних, як температура, час і характеристики сировини.

Подібна задача може вирішуватися таким чином:

1. **Система спостерігається і збираються дані для складання навчальної множини**. Кожен елемент навчальної множини складається з вимірів під час спостережень і включає значення всіх входів (вхідний вектор) і всіх виходів (вихідний вектор).
2. **Мережа навчається на цьому навчальній множині**. Навчання складається з пред'явлення вхідного вектора, обчислення вихідного вектора, порівнювання вихідного вектора з вхідним вектором, отриманим в процесі спостережень, і корекції ваг, що мінімізує різницю між ними. Кожен вхідний вектор пред'являється по черзі, і мережа частково навчається. Після великого числа передавальний вхідних векторів мережу зійдеться до рішення, яке мінімізує різницю між бажаними і виміряними виходами системи. Фактично мережа будує внутрішню модель невідомої системи. Якщо навчальна множина досить велике, мережа збігається до точної моделі системи. Якщо мережі пред'явити деякий вхідний вектор, відмінний від будь-якого з векторів, пред'явлених при навчанні, то повністю навчена мережа видасть той же самий вихідний вектор, що і справжня система.
3. **Максимізується цільова функція**. Цільова функція виходів повинна бути сконструйована таким чином, щоб висловлювати ступінь «задовільності» результату. Тепер входи стають змінними для навченої мережі. Вони підлаштовуються за допомогою того ж самого навчального алгоритму, який застосовувався для виставлення ваг на кроці 2, проте використовуються для максимізації цільової функції.

У багатьох випадках можуть бути присутніми обмеження, що накладаються завданням. Наприклад, може бути неможливо фізично брати значення змінних поза деякого діапазону. Ці обмеження (які можуть бути складними виразами) можуть бути легко враховані відкиданням на кроці 3 будь-якої зміни вхідної змінної, яке порушує обмеження.

Це узагальнення методу стохастичною оптимізації дозволяє його використовувати для широкого кола оптимізаційних задач. Можна застосовувати й інші методи, але стохастичний метод дозволяє подолати труднощі, обумовлені локальними мінімумами, з якими стикається метод зворотного поширення і інші методи градієнтного спуску. На жаль, імовірнісна природа процесу навчання може призводити до великого часу збіжності. Використання методів псевдотеплоемкості може істотно зменшити цей час, але процес все одно залишається повільним.

## **1.7 Зворотне поширення і навчання Коші**

Зворотне поширення має перевагу прямого пошуку, бо ваги завжди коригуються в напрямку, що мінімізує функцію помилки. Хоча час навчання і велике, воно істотно менше, ніж при випадковому пошуку, що виконується машиною Коші, коли знаходиться глобальний мінімум, але багато кроків виконуються в невірному напрямі, що віднімає багато часу.

Поєднання цих двох методів дало хороші результати. Корекція ваг, що дорівнює сумі, обчисленої алгоритмом зворотного поширення, і випадковий крок, задається алгоритмом Коші, призводять до системи, яка сходиться і знаходить глобальний мінімум швидше, ніж система, що навчається кожним з методів окремо. Проста евристика використовується для уникнення паралічу мережі, який може мати місце як при зворотному поширенні, так і при навчанні за методом Коші.

**Труднощі, пов'язані зі зворотним поширенням**

Незважаючи на міць, продемонстровану методом зворотного поширення, при його застосуванні виникає ряд труднощів, частина з яких, однак, полегшується завдяки використанню нового алгоритму.

**Збіжність**

В роботі доказ збіжності дається на мові диференціальних рівнянь в приватних похідних, що робить його справедливим лише в тому випадку, коли корекція ваг виконується за допомогою нескінченно малих кроків. Так як це веде до нескінченного часу збіжності, то воно втрачає силу в практичних застосуваннях. Насправді немає докази, що зворотне поширення буде сходитися при кінцевому розмірі кроку. Експерименти показують, що мережі зазвичай навчаються, але час навчання велике і непередбачувано.

**Локальні мінімуми**

У зворотному поширенні для корекції ваги мережі використовується градієнтний спуск, просувається до мінімуму відповідно до локальним нахилом поверхні помилки. Він добре працює в разі сильно порізаних неопуклих поверхонь, які зустрічаються в практичних завданнях. В одних випадках локальний мінімум є прийнятним рішенням, в інших випадках він неприйнятний.

Навіть після того як мережа навчена, неможливо сказати, чи знайдений за допомогою зворотного поширення глобальний мінімум. Якщо рішення незадовільно, доводиться давати ваг нові початкові випадкові значення і повторно навчати мережу без гарантії, що навчання закінчиться на цій спробі або що глобальний мінімум взагалі буде коли-небудь знайдений.

**Параліч**

При деяких умовах мережа може при навчанні потрапити в такий стан, коли модифікація ваг не веде до дійсних змін мережі. Такий «параліч мережі» є серйозною проблемою: один раз виникнувши, він може збільшити час навчання на кілька порядків.

Параліч виникає, коли значна частина нейронів отримує ваги, досить великі, щоб дати великі значення NET . Це призводить до того, що величина OUT наближається до свого граничного значення, а похідна від стискаючої функції наближається до нуля. Як ми бачили, алгоритм зворотного поширення при обчисленні величини зміни ваги використовує цю похідну в формулі як коефіцієнт. Для уражених паралічем нейронів близькість похідної до нуля призводить до того, що зміна ваги стає близьким до нуля.

Якщо подібні умови виникають у багатьох нейронах мережі, то навчання може сповільнитися до майже повної зупинки.

Ні теорії, здатної прогнозувати, чи буде мережа паралізована під час навчання чи ні. Експериментально встановлено, що малі розміри кроку рідше призводять до паралічу, але крок, малий для одного завдання, може виявитися великим для іншої. Ціна ж паралічу може бути високою. При моделюванні багато годин машинного часу можуть піти на те, щоб вийти з паралічу.

**Труднощі з алгоритмом навчання Коші**

Незважаючи на поліпшення швидкості навчання, що дається машиною Коші в порівнянні з машиною Больцмана, час збіжності все ще може в 100 разів перевищувати час для алгоритму зворотного поширення. Відзначимо, що мережевий параліч особливо небезпечний для алгоритму навчання Коші, особливо для мережі з нелінійністю типу логістичної функції. Нескінченна дисперсія розподілу Коші приводить до змін ваг необмеженої величини. Далі, великі зміни ваг будуть іноді прийматися навіть в тих випадках, коли вони несприятливі, часто приводячи до сильного насичення мережевих нейронів з випливають звідси ризиком паралічу.

**Комбінування зворотного поширення з навчанням Коші**

Корекція ваг в комбінованому алгоритмі, що використовує зворотне поширення і навчання Коші, складається з двох компонентів: (1) спрямованої компоненти, що обчислюється з використанням алгоритму зворотного поширення, і (2) випадкової компоненти, що визначається розподілом Коші.

Ці компоненти обчислюються для кожної ваги, і їх сума є величиною, на яку змінюється вага. Як і в алгоритмі Коші, після обчислення зміни ваги обчислюється цільова функція. Якщо має місце поліпшення, зміна зберігається. В іншому випадку воно зберігається з імовірністю, що визначається розподілом Больцмана.

Корекція ваги обчислюється з використанням представлених раніше рівнянь для кожного з алгоритмів:

(3)

де η - коефіцієнт, керуючий відносними величинами Коші і зворотного поширення в компонентах вагового кроку. Якщо η прирівнюється нулю, система стає повністю машиною Коші. Якщо η прирівнюється одиниці, система стає машиною зворотного поширення.

Зміна лише одного вагового коефіцієнта між обчисленнями вагової функції неефективно. Виявилося, що краще відразу змінювати все ваги цілого прошарку, хоча для деяких завдань може виявитися вигідніше інша стратегія.

Подолання мережевого паралічу комбінованим методом навчання. Як і в машині Коші, якщо зміна ваги погіршує цільову функцію, - за допомогою розподілу Больцмана вирішується, зберегти чи нове значення ваги або відновити попереднє значення. Таким чином, є кінцева ймовірність того, що погіршує безліч збільшень ваг буде збережено. Так як розподіл Коші має нескінченну дисперсію (діапазон зміни тангенса тягнеться від - ¥ до + ¥ на області визначення), то досить імовірно виникнення великих збільшень ваг, часто призводять до мережевого паралічу.

Очевидне рішення, що складається в обмеженні діапазону зміни вагових кроків, ставить питання про математичної коректності отриманого таким чином алгоритму. Доведено збіжність системи до глобального мінімуму лише для початкового алгоритму. Подібного докази при штучному обмеженні розміру кроку не існує. Насправді експериментально виявлені випадки, коли для реалізації деякої функції потрібні великі ваги, і два великих ваги, віднімаючи, дають малу різницю.

Для вирішення проблеми паралічу був знайдений метод, що не порушує досягнутого навчання. Насичені нейрони виявляються за допомогою вимірювання їх сигналів OUT . Коли величина OUT наближається до свого граничного значення, позитивному чи негативному, на ваги, що живлять цей нейрон, діє стискає функція. Вона подібна до використовуваної для отримання нейронного сигналу OUT , за винятком того, що діапазоном її зміни є інтервал (+5, -5) або інше відповідне безліч. Тоді модифіковані вагові значення рівні:

Ця функція сильно зменшує величину дуже великих ваг, вплив на малі ваги значно слабший. Далі вона підтримує симетрію, зберігаючи невеликі відмінності між великими вагами. Експериментально було показано, що ця функція виводить нейрони зі стану насичення без порушення досягнутого в мережі навчання. Не було витрачено серйозних зусиль для оптимізації використовуваної функції, інші значення констант можуть виявитися кращими.

**ВИСНОВКИ ПО ПЕРШОМУ РОЗДІЛУ**

Комбінована мережа, що використовує зворотне поширення і навчання Коші, навчається значно швидше, ніж кожен з алгоритмів окремо, і щодо нечутлива до величинам коефіцієнтів. Збіжність до глобального мінімуму гарантується алгоритмом Коші, в сотнях експериментів з навчання мережу жодного разу не потрапляла в пастки локальних мінімумів. Проблема мережевого паралічу була вирішена за допомогою алгоритму селективного стиснення ваг, який забезпечив збіжність у всіх пред'явлених тестових завданнях без істотного збільшення навчального часу.

Незважаючи на такі обнадійливі результати, метод ще не досліджений до кінця, особливо на великих завданнях.

# РОЗДІЛ 2

# ЧАСОВІ РЯДИ

## **2.1 Системи сучасного управління мікрокліматом**

Часовий ряд є рядом точок даних, проіндексованих (перелічених чи відкладених графічно) хронологічно. Найчастіше він є послідовністю, розміщеною на точках, які рівновіддалені в часі та йдуть поступово. Одже, він є послідовністю даних дискретного часу. Наприклад, часовим рядям є кількості сонячних плям або висоти океанських припливів. За допомогою лінійних діаграм зображують часові ряди. Вони застосовуються в статистиці, прогнозуванні погоди, розпізнаванні образів, передбаченні траєкторій та розумному транспорті, передбаченні землетрусів та в більшості областей прикладної науки, де використовуються часові вимірювання.[14]

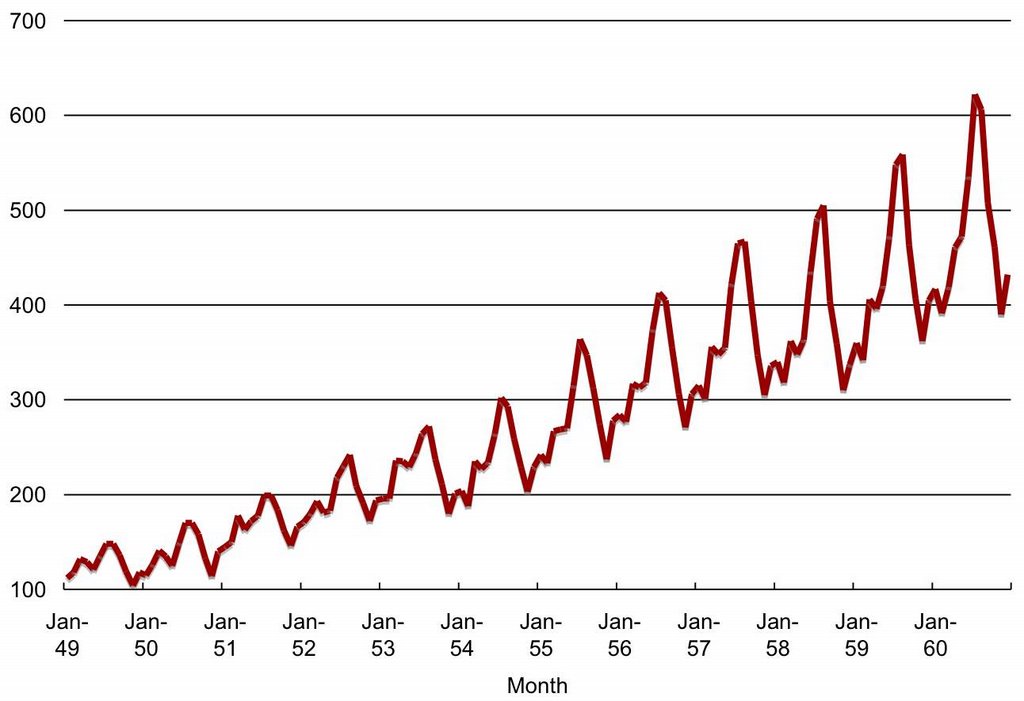


Рисунок 2.1 - Один із прикладів зображення часового ряду

Аналіз часових рядів є сукупністю математико-статистичних методів аналізу, що застосовується для обчислення структури часових рядів та їх прогнозування. Обрахунок побудови часового ряду необхідний для побудови математичної моделі явища, що є основою аналізованого ряду. Передбачення майбутнього показника використовується для того, щоб правильно прийняти рішення.

Для аналізу часових рядів застосовують методи аналізу часових рядів для того, щоб отримати значимістатистики та інші характеристики даних. Прогнозування часових рядів являє собою використання моделей для прогнозування майбутніх показників на основі значень попередньо отриманих. Застосування регресійного аналізу для перевірки теорій про поточні показники декількох незалежних часових рядів мають вплив на поточне значення різних часових рядів.

Природний часовий порядок є головним критерієм даних часових рядів. З цього випливає, що аналіз часових рядів є відмінним від попередніх, у яких відсутній природній порядок спостережень. Аналізуючи часові ряди помічаємо їх відмінність від аналізу просторових даних, у яких спостереження майже завжди відносяться до географічного місцеположення. Зазвичай, стохастична модель часового ряду показуватиме фактично те, що близькі в часі спостереження, будуть пов'язані тісніше, ніж ті, що знаходяться на дальній відстані одне від одного.

Аналіз часових рядів можна застосовувати до дійсно значних дискретних символьних даних, неперервних даних та дискретних числових даних.Методи аналізу часових рядів можна розділити на два класи: метод частотної області та метод часової області. Перший включає вейвлетний та спектральний аналізи, другий — взаємної кореляції та аналіз автокореляції. У часовій області кореляція й аналіз можуть здійснюватися фільтроподібним способом із використанням масштабної кореляції, що призведе до зменшення потреби діяти в частотній області.

Методики аналізу часових рядів можуть також поділятися на непараметричні та параметричні. Непараметричні підходи дають оцінку коваріації або спектру процесу без припущення про наявність у такого процесу якоїсь певної структури. На противагу цьому, параметричні підходи передбачають, що стаціонарний стохастичний процес, який є основою даних, має структуру, яку можна описати із застосуванням певного числа параметрів. В цих підходах задачею є оцінка параметрів моделі, що описує цей стохастичний процес. [15]

Існують певні цілі аналізу часових рядів:

* визначення природи ряду ;
* прогнозування (передбачення майбутніх показників часового ряду по отриманим значенням).

Ці цілі потребують ідентифікації моделей ряду і деякогоформального опису. Не зважаючи на глибоке розуміння і справедливість теорії, можна екстраполювати ряд на основі знайденої моделі, цебто спрогнозувати його майбутні значення.

Прогнозні оцінки за допомогою методів екстраполяції розраховуються в кілька етапів:

* перевірка базової лінії прогнозу;
* виявлення закономірностей минулого розвитку явища;
* оцінка ступеня достовірності виявленої закономірності розвитку явища в минулому (підбір трендової функції);
* [екстраполювання](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D1%82%D1%80%D0%B0%D0%BF%D0%BE%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) — перенесення виявлених закономірностей на деякий період майбутнього;
* коректування отриманого прогнозу з урахуванням результатів змістовного аналізу поточного стану.

Для отримання об'єктивного прогнозу розвитку досліджуваного явища дані базової лінії повинні відповідати таким вимогам:

* крок за часом для всієї базової лінії повинен бути однаковий;
* спостереження фіксуються в один і той же момент кожного часового відрізку;
* базова лінія повинна бути повною, тобто пропуск даних не допускається.

Для часових рядів існує кілька типів задач і типів аналізу даних, які підходять для різних цілей тощо. [16]

## **2.2 Прості моделі прогнозування**

При створення простих моделей передбачається, що деяка головна стадія прогнозованого тимчасового ряду доцільніше описує майбутнє спрогнозованого ряду, з цього слідує, що в цих моделях прогноз зазвичай дуже простий функцією від значень Прогнозоване змінної в недалекому минулому.

Найпростішою моделлю є

(5)

що відповідає припущенню, що "завтра буде як сьогодні".

При створення простих моделей передбачається, що деяка головна стадія прогнозованого тимчасового ряду доцільніше описує майбутнє спрогнозованого ряду, з цього слідує, що в цих моделях прогноз зазвичай дуже простий функцією від значень Прогнозоване змінної в недалекому минулому. Без сумніву, від такої примітивної моделі не слід чекати великої точності. Вона і не враховує механізми, які визначають прогнозовані дані (даний важливий недолік взагалі притаманний багатьом статистичним методам прогнозування), і не захищена від випадкових коливань, вона не враховує сезонні коливання і тенденції. Втім, можна будувати "наївні" моделі дещо по-іншому

, (6)

, (7)

такими способами ми намагаємося пристосувати модель до можливих тенденцій

, (8)

це спроба врахувати сезонні коливання.

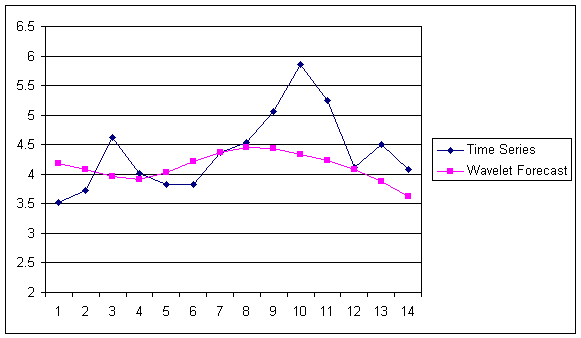


Рисунок 2.2 - Прогнозування найпростішими методами.

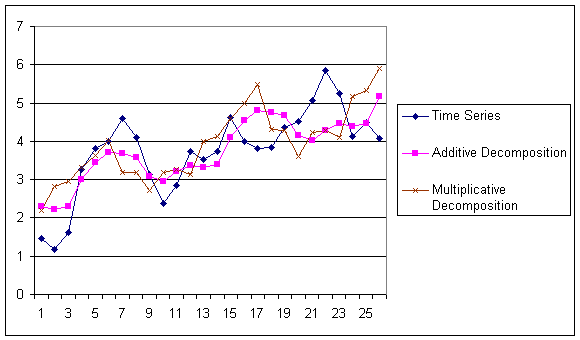
[](https://wiki.tntu.edu.ua/%D0%A4%D0%B0%D0%B9%D0%BB:24.png)

Рисунок 2.3 - Прогнозування найпростішими методами.

## **2.3 Середні та ковзаючи середні**

Найпростішою моделлю, заснованою на простому усереднюванні є

на відміну від найпростішої "наївної" моделі, яка підпорядковується принципу "завтра буде так, як сьогодні", ця модель підпорядковується принципу "завтра буде так, як було в середньому за останній час". Така модель стійкіша до коливань, оскільки в ній спрощуються випадкові викиди щодо середнього. Не зважаючи на це, даний метод ідеологічно настільки ж примітивний як і прості моделі і йому властиві майже ті ж самі недоліки.

У попередньо наведеній формулі прогнозувалося, що ряд усереднюється по досить тривалому інтервалу часу. Проте як правило, значення тимчасового ряду з недалекого минулого краще описують прогноз, ніж усі попередні значення цього ж ряду. Тоді можна використовувати для прогнозування ковзне середнє

Його значення полягає в тому, що модель помічає тільки те, що було недавно (на T відліків за часом в глибину) і ґрунтуючись тільки на цих даних будує прогноз.

При прогнозуванні досить часто використовується метод експоненціальних середніх, який постійно адаптується до даних шляхом нових значень. Формула, що описує цю модель записується як

де – прогноз на наступний період часу; – реальне значення у момент часу t; – минулий прогноз на момент часу t; – постійна згладжування (0 <= <= <= 1)

У цьому методі є внутрішній параметр а, який визначає залежність прогнозу від усіх розглянутих даних, причому вплив даних на прогноз експоненціально зменшується із "віком" даних. Залежність впливу даних на прогноз при різних коефіцієнтах а приведена на графіці.

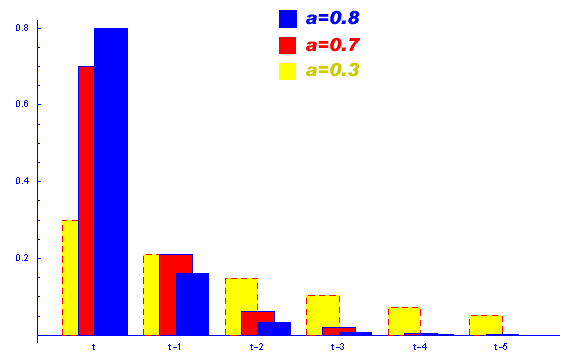


Рисунок 2.4 - Залежність впливу даних на прогноз при різних коефіцієнтах

## **2.4 Методи Хольта та Брауна**

В середині 20-го століття Хольтом було запропоновано метод експоненціального згладжування, який пізніше назвали на його честь. У цьому алгоритмі значення рівня і тенденції згладжуються методом експоненціального згладжування. При цьому у них різні параметри згладжування

Згладжений ряд загального рівня описує перше рівняння. Друге рівняння служить для того, щоб оцінити тренд. У третьому рівнянні визначається прогнозування на ρ відліків вперед за часом.

В методі Хольта сталі згладжування ідеологічно виконують ту ж функцію, що і стала в нескладному експоненціальному згладжуванні. Вони можуть підбиратися шляхом перебору по цих параметрах з якимсь кроком. Можна використовувати і менш складні в сенсі кількості обчислень алгоритми. Головне, що завжди можна підібрати таку пару параметрів, яка дає велику точність моделі на тестовому наборі і потім використовувати цю пару параметрів при реальному прогнозуванні.

Окремим випадком методу Хольта є метод Брауна, коли α = β.

## **2.5 Метод Вінтерса**

Хоча описаний вище метод Хольта (метод двопараметричного експоненціального згладжування) і не є зовсім простим (щодо "наївних" моделей і моделей, заснованих на усередненні), він не дозволяє враховувати сезонні коливання при прогнозуванні. Говорячи більш акуратно, цей метод не може їх "бачити" в передісторії. Існує розширення методу Хольта до трипараметричного експоненціального згладжування. Цей алгоритм називається методом Вінтерса. При цьому робиться спроба врахувати сезонні складові в даних. Система рівнянь, що описують метод Вінтерса виглядає наступним чином:

Дріб в першому рівнянні служить для виключення сезонності з . Після виключення сезонності алгоритм працює з "чистими" даними, в яких немає сезонних коливань. З'являються вони вже в самому фінальному прогнозі, коли "чистий" прогноз, порахований майже по методу Хольта множиться на сезонний коефіцієнт.[17]

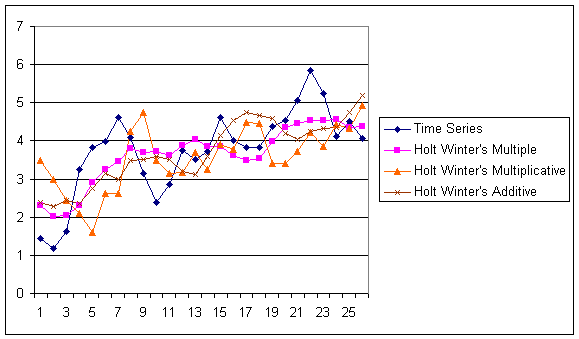
[](https://wiki.tntu.edu.ua/%D0%A4%D0%B0%D0%B9%D0%BB:25.png)

Рисунок 2.5 - Прогнозування методи Вінтерса.

## **2.6 Регресійні методи прогнозування**

Паралельно з методами, описаними вище, які засновуються на експоненціальному згладжуванні, вже певний проміжок часу для того, щоб спрогнозувати використовується регресійні алгоритми. Головною ідеєю алгоритмів даного класу можна описати наступним чином: існує прогнозована залежна змінна Y і відібраний заздалегідь набір змінних, від яких вона залежить, - , ..., (незалежні змінні). Походження незалежних змінних буває різним. Припустимо, що Y - рівень попиту на певний продукт на наступному тижні, то незалежними змінними можуть бути рівень попиту на цей же продукт в минулий і позаминулий тижні, рівень платоспроможності населення, витрати на рекламу, діяльність конкурентів, економічна обстановка і багато іншого. Головне - уміти формалізувати всі зовнішні чинники, від яких може залежати рівень попиту в числовій формі.

У простішому варіанті лінійної регресійної моделі залежність залежної змінної від незалежних має вигляд:

Тут , ,,…, - підбирані коефіцієнти регресії; - компонента помилки.

Передбачається, що всі помилки незалежні і нормально розподілені. Для побудови регресійних моделей необхідно мати базу даних спостережень приблизно такого вигляду:

Таблиця 2.1 – База даних спостережень

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Змінні** | | | | | |
| **Незалежні** | | | | | **Залежна** |
| № |  |  | … |  |  |
| 1 |  |  | … |  |  |
| 2 |  |  | … |  |  |
| … | … | … | … | … | … |
| m |  |  | … |  |  |

За допомогою таблиці значень минулих спостережень можна підібрати (наприклад, методом найменших квадратів) коефіцієнти регресії, побудувавши тим самим модель.

При роботі з регресією треба дотримуватися певної обережності і обов'язково перевірити на адекватність знайдені моделі. Існують різні способи такої перевірки. Обов'язковим є статистичний аналіз залишків, тест Дарбіна-Уотсона. Корисно, як і у випадку з нейронними мережами, мати незалежний набір прикладів, на яких можна перевірити якість роботи моделі.[18]

**ВИСНОВКИ ПО ДРУГОМУ РОЗДІЛУ**

Наведені методи у другому розділі показують, що технології нейронних мереж і генетичних алгоритмів застосовані практично в будь-якій області. У деяких завданнях, таких як прогнозування котирувань або розпізнавання образів, нейромережі стали вже звичним інструментом. Немає сумнівів, що повсюдне проникнення нових технологій і в інші області - тільки питання часу. Аналогічний процес уже стався з комп'ютерами, які протягом всього 20 років поширилися з військових обчислювальних центрів в кожен офіс і в багато квартири.

Впровадження нових наукоємних технологій в комерційній фірмі - досить непроста справа, що вимагає, крім грошей і часу, ще й деякої зміни психології. Однак, практика показує, що ці вкладення окупляться і виводять компанію на якісно новий рівень.

# РОЗДІЛ 3

# ОПИС ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ НЕЙРОМОДЕЛІ КОНТРОЛЮ ВАРТОСТІ ВІДЕОАПАРАТУРИ

Для рішення задачі, що поставлено, використаємо програму Statistica, а саме пакет Neural Networks. Нейронні мережі успішно застосовуються в різних галузях, наприклад, бізнесі, медицині, техніці, геології, фізики. Вони увійшли в практику скрізь, де потрібно вирішувати задачі класифікації, прогнозування або управління. Такий неймовірний успіх визначається декількома причинами.

По-перше – це багато можливостей. Нейронні мережі – це виключно потужний метод моделювання, що дозволяє втілювати надзвичайно складні залежності. Зокрема, нейронні мережі нелінійні по своїй природі. Лінійне моделювання було основним методом моделювання в більшості областей протягом багатьох років, оскільки для нього добре розроблені процедури оптимізації. Лінійні моделі працюють погано у завданнях, де лінійна апроксимація незадовільна (а таких досить багато). Крім того, нейронні мережі справляються з "прокляттям розмірності", яке не дозволяє моделювати лінійні залежності у випадку великого числа змінних.

Друге – це простота у використанні. Нейронні мережі навчаються на прикладах. Користувач нейронної мережі підбирає представницькі дані, а потім запускає алгоритм навчання, який автоматично сприймає структуру даних. При цьому від користувача, звичайно, потрібно якийсь набір евристичних знань про те, як саме слід відбирати та готувати дані, обирати потрібну архітектуру мережі й інтерпретувати результати, проте рівень знань, необхідний для успішного застосування нейронних мереж, набагато скромніше, ніж, наприклад, при використанні традиційних методів статистики.

Нейронні мережі привабливі з інтуїтивної точки зору, бо вони засновані на примітивній біологічній моделі нервових систем. Розвиток таких нейро-біологічних моделей може призвести до створення дійсно мислячих комп'ютерів у майбутньому. Тим часом уже "прості" нейронні мережі, що будує система ST Neural Networks, є потужною зброєю в арсеналі фахівця з прикладної статистики.

## **3.1 Рішення задач класифікації**

У задачі класифікації мережа повинна віднести кожне спостереження до одного з декількох класів (або, в більш загальному випадку, оцінити ймовірність приналежності спостереження до кожного з класів). У пакеті ST Neural Networks для класифікації використовується номінальна вихідна змінна – різні її значення відповідають різним класам.

У пакеті ST Neural Networks класифікацію можна здійснювати за допомогою мереж наступних типів: багатошарового перцептрона, радіальної базисної функції, ймовірнісної нейронної мережі і лінійної мережі.

Номінальні змінні представляються в пакеті ST Neural Networks в одному з двох видів (перший з них підходить тільки для змінних з двома значеннями):

* бінарному (два стани) ;
* один-з-N.

При бінарному представленні змінної відповідає один вузол мережі, при цьому значення 0.0 означає активний стан, а 1.0 – неактивний. При кодуванні 1-з-N на кожний стан виділяється один елемент, так що кожний конкретний стан представляється як 1.0 у відповідному елементі і 0.0 у всіх інших.

Номінальні вхідні змінні в пакеті ST Neural Networks можуть бути перетворені одним з цих методів як на етапі навчання мережі, так і при її роботі. Цільові вихідні значення для елементів, відповідних номінальною змінною, також легко визначаються під час навчання. Дещо більші зусилля потрібні на те, щоб за результатами роботи мережі визначити вихідний клас.

Кожен з вихідних елементів буде містити числові значення в інтервалі від 0.0 до 1.0. Щоб впевнено визначити клас по набору вихідних значень, мережа повинна вирішити, "чи достатньо близькі" вони до нуля або одиниці. Якщо такої близькості не спостерігається, клас вважається "невизначеним".

Крім того, в пакеті ST Neural Networks для інтерпретації вихідних значень використовуються довірчі рівні (пороги і відкидання). Ці порогові значення можна коригувати, щоб змусити мережу бути більш або, навпаки, менш "рішучою" при оголошенні класу. Схеми тут небагато розрізняються для випадків бінарного і 1-з-N уявлень:

Бінарне. Якщо вихідне значення елемента перевищує поріг прийняття, то вибирається клас 1.0. Якщо вихідне значення лежить нижче порога відкидання, вибирається клас 0.0. Якщо вихідне значення лежить між порогами, клас вважається не визначеним.

Один – через N. Певний клас вибирається тільки в тому випадку, коли значення відповідного вихідного елемента вище порога прийняття, а в усіх інших вихідних елементів – нижче порога відкидання. Якщо ж дана умова не виконана, клас не визначається.

Ось точна схема дії алгоритму визначення класу в пакеті ST Neural Networks. Спочатку обирається елемент з найвищим вихідним сигналом. Якщо його вихідний сигнал вище або дорівнює порогу прийняття, а вихідні сигнали всіх інших елементів нижче порога відкидання, то в якості відповіді видати клас, який визначається цим елементом.

Все сказане відноситься до механізму вибору класу для більшості типів мереж: MLP, RBF, лінійних мереж і PNN (один виняток: в PNN-мережі не можна використовувати бінарне кодування, і навіть бінарні номінальні вихідні змінні оцінюються за допомогою кодування 1-из-N).

У наших розглядах ми припускали, що "позитивного" рішенням про класифікацію має відповідати значення, близьке до 1.0, а "негативного" – близьке до 0.0. Це дійсно так в тому випадку, якщо на виході використовуються логістичні функції активації. Крім того, це зручно, оскільки ймовірність може приймати значення від 0.0 до 1.0. Проте, в деяких ситуаціях може виявитися більш зручним використовувати інший діапазон. Іноді застосовується зворотна упорядкованість, так що позитивне рішення відповідає малим вихідним значенням. Пакет ST Neural Networks підтримує будь-який з цих варіантів роботи.

## **3.2 Моделювання етапів розробки ЕС для заданої предметної області.**

***Етап ідентифікації***

Основним завданням є створення ЕС для автоматичного визначення ціни на фотокамеру за певними ознаками. Цілью розробки є можливість автоматичного визначення ціна на фотокамеру для досягнення найбільшого успіху.

У створенні подібних експертних систем беруть участь, як правило, математик, програміст та бізнес-аналітик. Основна роль в розробці такої системи належить бізнес-аналітику. Основні користувачі даної ЕС - компанії, що виробляють фото обладнання та їх продукт-менеджери.

***Етап концептуалізації***

На цьому етапі проводиться змістовний аналіз проблемної області та відбувається визначення характеристик фотокамери, що можуть впливати на її ціну. Потрібно обрати ті характеристики, які присутні у всіх камерах, та можуть впливати на її ціну.

В даному прикладі було обрано наступні характеристики: виробник, дата випуску, максимальна та мінімальна роздільна здатність, кількість ефективних пікселів, збільшення ширококутного та теле об'єктивів, нормальна та макро фокусна відстань, розмір накопичувача, вага та розміри.

Збір інформації відбувається шляхом аналізу ринку та опитування експертів.

***Етап формалізації***

На етапі формалізації було обрано інструментальним засобом програмний комплекс STATISTICA 10.0 і визначино способи подання всіх видів знань, а саме: максимальна та мінімальна роздільна здатність в кількості пікселів. Збільшення ширококутного та теле об'єктивів, нормальна та макро фокусна відстань в міліметрах. Наступні ознаки будуть подаватися на вхід системи у такий спосіб: Розмір внутрішнього накопичувача – у мегабайтах, вага – в грамах, розміри – в міліметрах.

Загальний принцип побудови експертної системи визначення ціни, це аналіз ринку та зібрання достовірної інформації.

***Етап виконання***

Мета цього етапу – створення одного або декількох прототипів ЕС, вирішуючих необхідні завдання. Розробка прототипу полягає в програмуванні його компонентів або виборі їх з відомих інструментальних засобів і наповненні бази знань.

***Етап тестування***

В ході етапу тестування проводиться оцінка обраного способу представлення знань в ЕС в цілому. Для цього підбираються приклади, що забезпечують перевірку всіх можливостей нової ЕС, тестується якість і коректність інформації, яка видається ЕС на виході, повноту бази знань і інші фактори роботи системи.

## **3.3 Постановка задачі.**

Дана вибірка фотокамери. Вона містить в собі 10 стовпців вхідних даних:

* Модель
* Дата випуску
* Максимальна роздільна здатність
* Мінімальна роздільна здатність
* Ефективні пікселі
* Ширококутний зум
* Зум телеоб’єктива
* Нормальна фокусна відстань
* Макрофокусна відстань
* Розмір накопичувача
* Вага
* Розмір

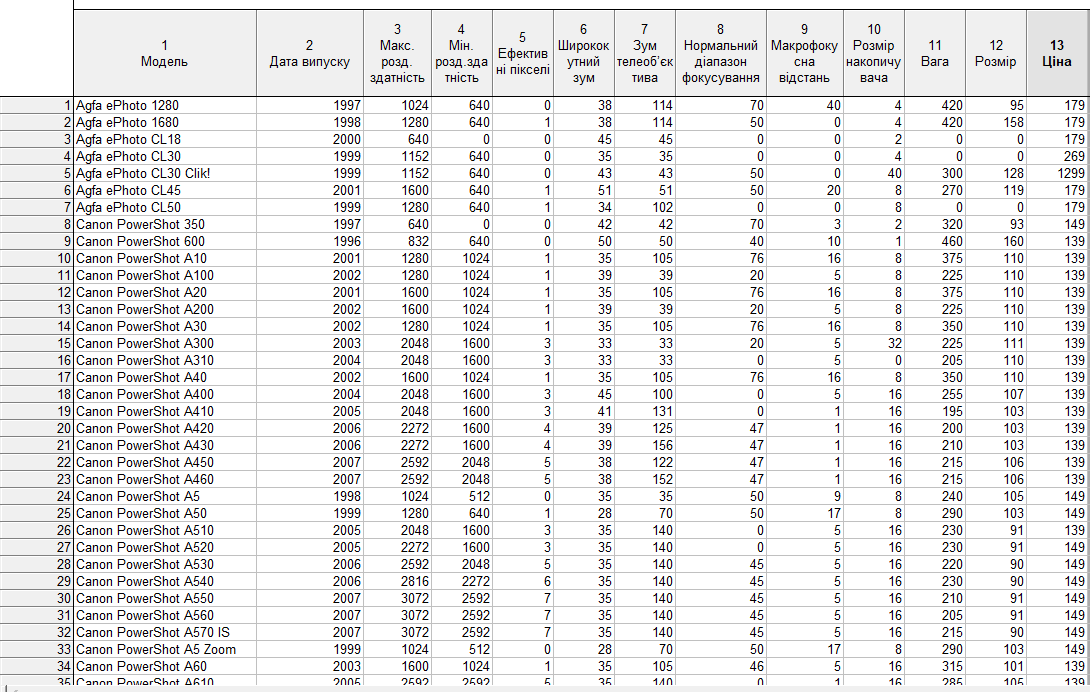
Стовбець вихідних даних - ціна. Він визначається заданими вище факторами та вимірються в долларах США.

Задано 1000 спостережень.

Необхідно розробити модель класифікатора, досягнувши критерію мінімальних помилок та максимальної продуктивності, а також досягти правильної класифікації на тестових даних.

## **3.4 Розв’язання задачі.**

Для розв’язання поставленої задачі запускаємо вибірку у програмному середовищі STATISTICA 10.0. На Рис. 3.1 зображено фрагмент навчальної вибірки.

 Рисунок 3.1 - Фрагмент навчальної вибірки

Запускаємо модуль Нейронні мережі STATISTICA. На стартовому вікні вибираємо тип завдання: *Анализ.* За інструмент обираємо *Нейронные сети.*

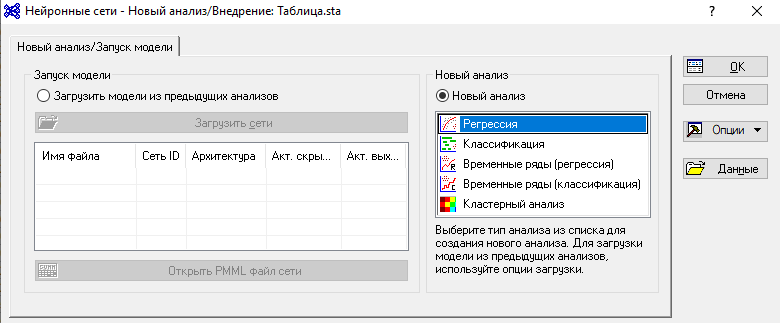


Рисунок 3.2 - Вікно *Нейронные сети*

Далі, переходимо до задання змінних. Для цього на стартовому вікні модуля, на вкладці *Быстрый*, натискаємо кнопку *Переменные*.

У діалоговому вікні вибираємо змінні, в даному прикладі є вихідна (залежна) змінна і 12 вхідних змінних. У лівому вікні виберемо безперервну вихідну змінну, - це змінна «Ціна» з номером 13. Виберемо вхідні змінні - це незалежні змінні, які передбачають відгук.

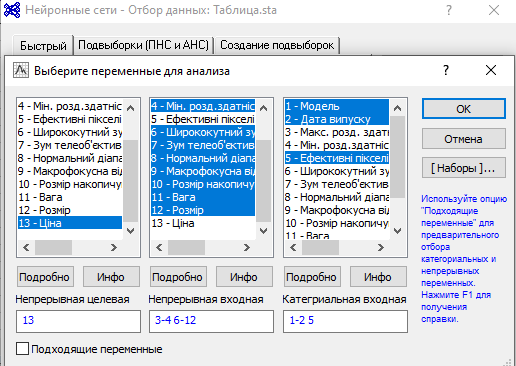


Рисунок 3.3 - Вікно задання змінних

Далі на вкладці *Быстрый* обираємо *Сети для обучения – 20, сети для сохранения* – 10.

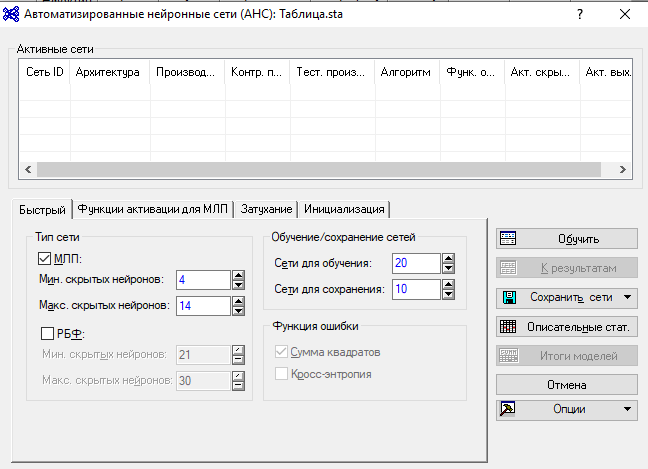


Рисунок 3.4 – Вікно обрання кількості мереж.

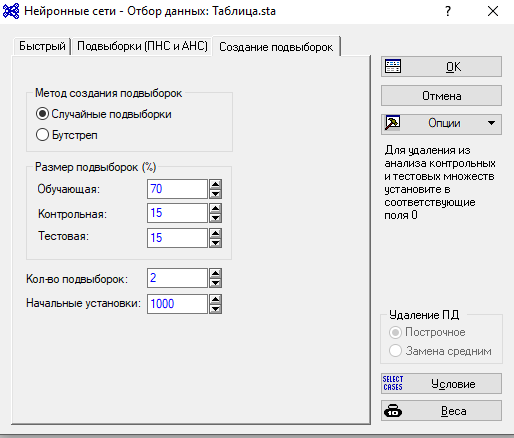


Рисунок 3.5 - Вікно *Отбор данных*

Розміри вибірок обираємо: Навчальна – 70%, Контрольна – 15%, Тестова– 15%.

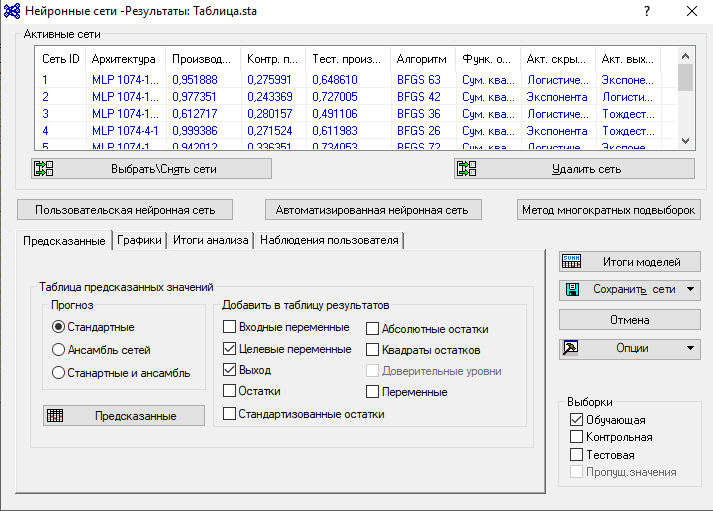


Рисунок 3.6 - Результати навчання мережі

Отже, отримали 10 найкращих мереж. На кладці *Быстрый* обираємо *Итоги сетей*. Бачимо докладний результат.

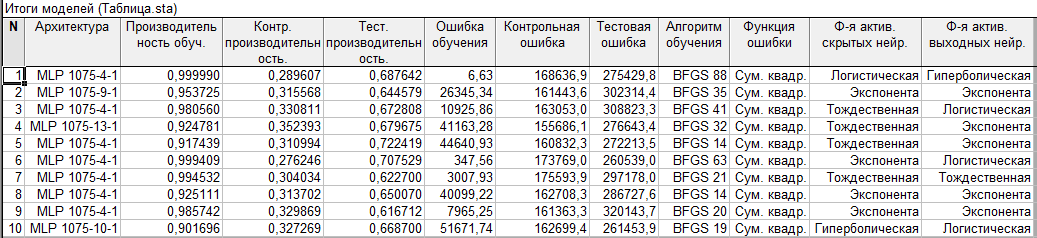


Рисунок 3.7 - Докладні результати моделей

Отже, мережа MLP 1075-4-1дала найкращий результат, а саме:

* Продуктивність – 0,99999028.
* Найменший діапазон помилок.

Зберігаємо модель.

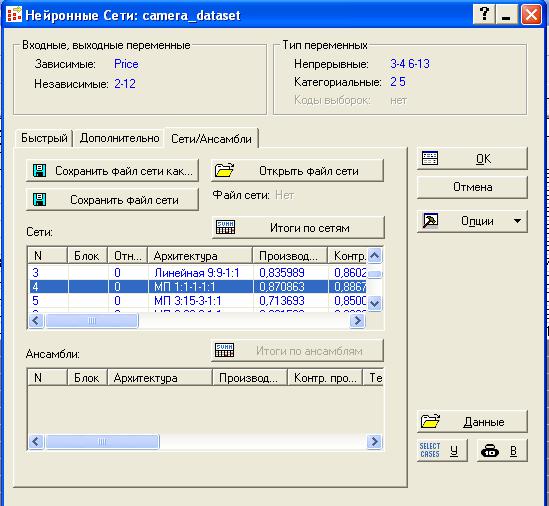


Рисунок 3.8 - Збереження моделі

Наступним кроком буде тестування навченої моделі на тестових даних. Отже, у вікні *Нейронные сети* вкладка *Быстрый* обираємо змінні, а у вкладці *Сети/Ансамбли* натискаємо *Открыть файл сети* та знаходимо збереженумодель.

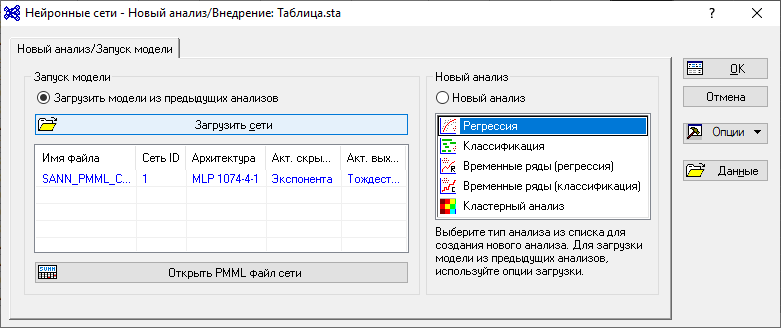


Рисунок 3.9 - Вікно *Нейронные сети*

У вкладці *Дополнительно* обираємо *Запустить модель.*

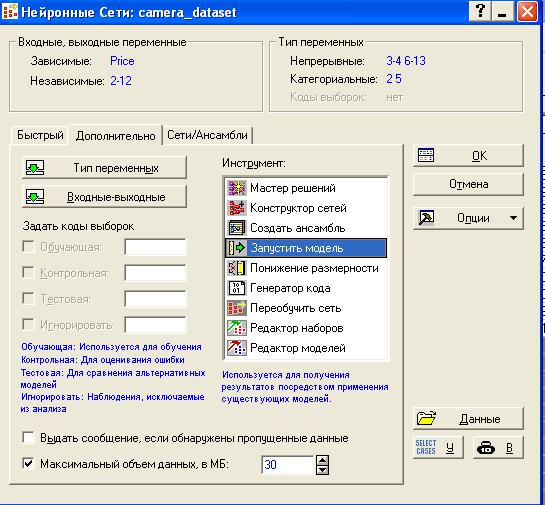


Рисунок 3.10 - Вікно *Нейронные сети*

Наступним кроком відкриваємо вкладку *Наблюдения пользователя*.

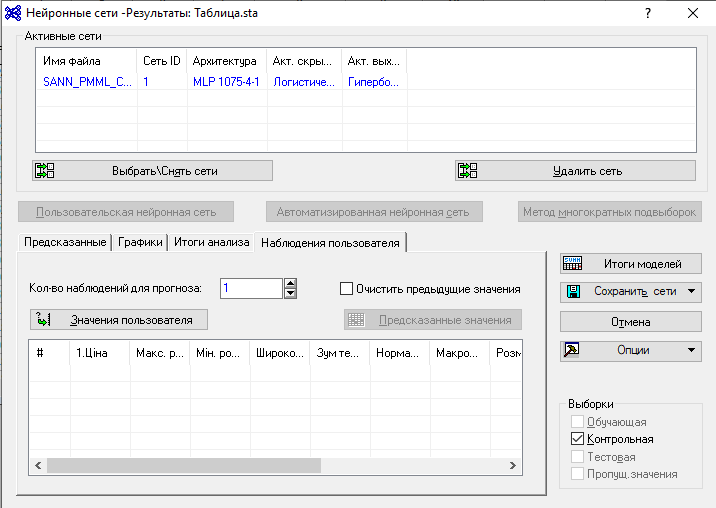


Рисунок 3.11 - Результати мережі

Натискаємо *Значения пользователя* та вводимо дані, котрі потрібно класифікувати.

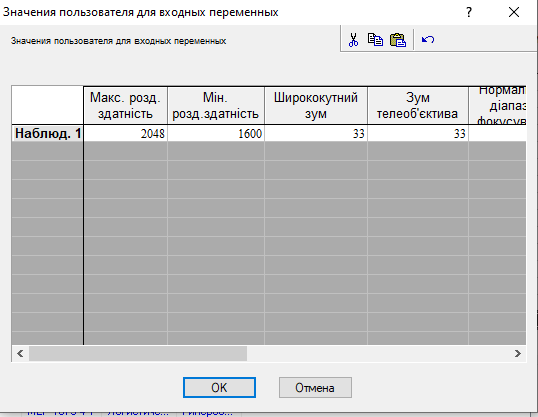


Рисунок 3.12 - Вікно *Значения пользователя*

Натиснувши ОК, повертаємося до вікна *Наблюдения пользователя* та нитаскаємо *Предсказанные*. Отримали результат:

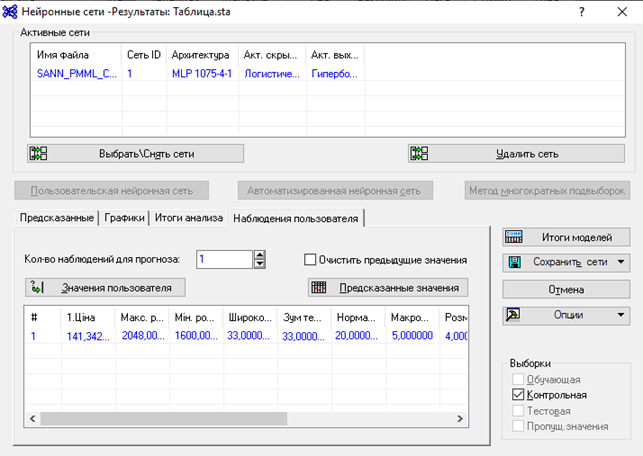


Рисунок 3.13 - Вікно *Прогноз наблюдения пользователя*

Отже, система автоматично визначила ціну в 141$, що відповідає дійсності (ціна для схожої камери з вибірки 139$). Для перевірки проведемо ще один прогін.



Рисунок 3.14 - Вікно *Прогноз наблюдений пользователя для камери*

Casio QV-4000

Система автоматично визначила ціну в 250$, що відповідає дійсності (ціна для схожої камери з вибірки 249$).



Рисунок 3.15 - Вікно *Прогноз наблюдений пользователя для камери*

Kodak DX3900

Система автоматично визначила ціну в 132$, що відповідає дійсності (ціна для схожої камери з вибірки 129$).

Отже, розглянувши декілька прикладів на нових даних, можна сказати, що система прогнозую дані вірно, а отже, модель дійсно гарно навчилася та може бути використана для визначення ціни для нових камер.

**ВИСНОВКИ ПО ТРЕТЬОМУ РОЗДІЛУ**

У третьому розділі було розглянуто етапи проектування експертних систем з практичним обгрунтуванням відповідно до заданої предметної області – визначення ціни фотокамери за ознаками, котрі визначили експерти.

У третьому розділі було проведено аналіз для навчальної вибірки та за допомогою майстра рішень знайдено кращу модель:

● Продуктивність – 0,99999028.

Після збереження навченої моделі було проведено перевірку на правильність лінійної регресії за ознаками. Інструментарії програмного засобу STATISTICA дозволяють це робити досить зручно. Для цього запускається збережена модель, обирається *Наблюдения пользователя* та у вікні *Значения пользователя* вводяться дані, яким треба назвачити клас. Значення, отримані мережею, відповідають очікуванням, а отже, це говорить про те, що модель добре навчилась і є оптимальною для розв’язання поставленої задачі.

Отже, в ході виконання роботи було створено експертну систему для автоматичного визначення ціни фотокамери за певними ознаками.

# ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було створено інтелектуальну нейромодель контролю вартості відеоапаратури.

Результат даного дослідження полягає в тому, що проблема оцінки інформативності простору ознак може бути вирішена за допомогою нейромережевих технологій.

Під час виконання даної роботи були вирішені наступні задачі:

* дослідити недоліки існуючих нейромережевих технологій;
* знайти підходи до виправлення недоліків існуючих нейромережевих технологій, не втрачаючи при цьому продуктивність;
* проаналізувати різноманіття ознак при діагностиці вартості відеоапаратури та проблеми оцінки їх інформативності;
* розробити на основі вивченого матеріалу алгоритм для реалізації технології;
* сформувати набір навчальних даних;
* створити ансамбль нейромережевих технологій, що задовольняють задачу визначення стану самопочуття організму;
* провести аналіз отриманих результатів.

Для проектування інтелектуальної нейромоделі контролю вартості відеоапаратури було використано основні показники, які використовують для визначення вартості відеоапаратури, такі як виробник, модель, дата випуску, максимальна роздільна здатність, мінімальна роздільна здатність, ефективні пікселі, ширококутний зум, зум телеоб’єктива, нормальна фокусна відстань, макрофокусна відстань, розмір накопичувача, вага та розмір.

У дипломній роботі було розглянуто етапи проектування експертних систем з практичним обгрунтуванням відповідно до заданої предметної області – визначення ціни фотокамери за ознаками, котрі визначили експерти; було проведено аналіз для навчальної вибірки та за допомогою майстра рішень знайдено кращу модель:

● Продуктивність – 0,99999028

Під час виконання кваліфікаційної роботи, було досягнено поставленої мети, а саме: дослідження нейромережевих технологій при діагностиці вартості відеоапаратури та розробка апаратно-програмних засобів інтелектуальної нейромоделі контролю вартості відеоапаратури, з можливістю маcштабування та додавання інших параметрів. Розроблене нейромережеве забезпечення задовольняє всім вимогам, поставленим на етапі постановки завдання.

# ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Базові поняття ШІ [Електронний ресурс] // UAdmin. – 2012. – Режим доступу до ресурсу: <https://knhelp.wordpress.com/2012/04/19/л1-базові-поняття-штучного-інтелект/>.
2. Романов В.П. Інтелектуальні інформаційні системи в економіці [Електронний ресурс] / Романов В.П. // Видавництво «Іспит». – 2003. – Режим доступу до ресурсу: <https://ukrdoc.com.ua/text/39770/index-1.html>.
3. Про штучний інтелект [Електронний ресурс] – Режим доступу до ресурсу: <http://um.co.ua/6/6-2/6-21122.html>.
4. Особливості та сфера застосування експертних систем [Електронний ресурс] – Режим доступу до ресурсу: <http://www.virtual.ks.ua/essays-term-papers-and-diplomas/1316-features-and-scope-of-expert-systems.html>.
5. Логічний висновок в експертних системах [Електронний ресурс] – Режим доступу до ресурсу: <https://uadoc.zavantag.com/text/28522/index-1.html>.
6. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ: Загальні питання теорії проектування ІС [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://ela.kpi.ua/bitstream/123456789/33651/1/PIS_KL.pdf>.
7. Прогнозування котирування акцій з використанням нейромереж [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <http://mses.kpi.ua/diplom/bar.pdf>.
8. Стохастичні методи [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bestreferat.ru/referat-244575.html>.
9. Структура інтелектуальної системи [Електронний ресурс] – Режим доступу до ресурсу: <https://uadoc.zavantag.com/text/307/index-7.html>.
10. Область застосування експертних систем [Електронний ресурс] – Режим доступу до ресурсу: <http://um.co.ua/1/1-7/1-75776.html>.
11. Сілаенков О.М. Комп'ютерні системи підтримки прийняття рішень [Електронний ресурс] / Сілаенков О.М.. – 2007. – Режим доступу до ресурсу: <https://ukrdoc.com.ua/text/34550/index-1.html?page=2>.
12. Метод зворотного поширення помилки [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Метод_зворотного_поширення_помилки>.
13. ЕКСПЕРТНІ СИСТЕМИ Частина 1 [Електронний ресурс] / Месюра В. І., Яровий А. А., Арсенюк І. Р. // Вінниця ВНТУ. – 2006. – Режим доступу до ресурсу: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiLsJXO85TxAhXW3YUKHfboDjgQFjAAegQIAxAD&url=https%3A%2F%2Fiq.vntu.edu.ua%2Ffdb%2F896%2FYarovyy_EC_%25D0%259D%25D0%259F-2.doc&usg=AOvVaw0dr8efww7PAqVtS93Ywi4->.
14. Часовий ряд [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Часовий_ряд#:~:text=Часовий%20ряд%20(англ.,є%20послідовністю%20даних%20дискретного%20часу>.
15. Часовий ряд [Електронний ресурс] – Режим доступу до ресурсу: <https://wikizero.com/uk/Часовий_ряд>.
16. Аналіз часових рядів [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Аналіз_часових_рядів>.
17. Задачі прогнозування [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <https://studfile.net/preview/9774505/page:2/>.
18. Психологія і соціологія розглядають спонукання як поведінкове виявлення потреб, сконцентрованих на досягненні цілей [Електронний ресурс] – Режим доступу до ресурсу: <https://studopedia.org/4-97226.html>.
19. Карвацький С. С. Евристичний метод тренування штучної нейронної мережі : дис. канд. / Карвацький Сергій Сергійович – Київ, 2018. – 87 с.
20. Методи прогнозування [Електронний ресурс] – Режим доступу до ресурсу: <https://wiki.tntu.edu.ua/Методи_прогнозування>.
21. Інформаційна модель [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Інформаційна_модель>.
22. Марченко О.О. Навчально-методичний посібник. Актуальні проблеми Data Mining [Електронний ресурс] / Марченко О.О., Россада Т.В.. – 2017. – Режим доступу до ресурсу: <http://csc.knu.ua/media/filer_public/38/03/3803002b-e068-4a08-8a6c-a4edc183892a/datamining20170917.pdf>.
23. Альошин С.П. Навчальний посібник з дисципліни «Інтелектуальний аналіз даних» для студентів денної та заочної форми навчання: 6.050101 – «Комп’ютерні науки» / С. Альошин, О. Бородіна – Полтава: ПолтНТУ, 2014. – 187 с.
24. Вінстон П. Штучний інтелект / П. Вінстон. – Москва: Мир, 1978. – 520 с.
25. Морозов А.А. Ситуационное управление и системы поддержки принятия решений / А.А. Морозов, Г.С. Теслер // Збірник доповідей науково-практичної конференції з міжнародною участю „Системи підтримки прийняття рішень. Теорія і практика”. – Київ, 2005. – С. 5 – 9.
26. Морозов А.А. Состояние и перспективы нейросетевого моделирования СППР в сложных социотехнических системах / В.П. Клименко, А.Л. Ляхов, С.П. Алёшин // Математичні машини і системи. – 2010. - № 1.- С. 127 – 149.
27. Нейронні мережі. STATISTICA Neural Networks: Методологія і технології сучасного аналізу даних / за редакцією В. П. Боровикова. - 2-е вид. , перероб. і дод. - М.: Гаряча лінія-Телеком, 2008. - 392 с., мул.

# ДОДАТОК А

# Макрос аналізу вартості фотокамер

#include <stdio.h>

#include <math.h>

#include <string.h>

#include <stdlib.h>

#ifndef FALSE

#define FALSE 0

#define TRUE 1

#endif

#define MENUCODE -999

static double NNCode01Thresholds[] =

{

/\* layer 1 \*/

-0.36211287648326584, -0.16959220447501724, -0.052056294423798728, 0.029025989433569998,

1.1203095599118227, -0.021150679025728012,

/\* layer 2 \*/

0.64937647851333313, -0.48249171754056408, 0.55268474297823778, -0.70070551621911681,

-0.67877302040780263, -0.33717680625734009,

/\* layer 3 \*/

-0.45389533906546153, -2.1215019942767821, -0.036003182024363242, 0.95153455448844837,

-0.73833722317416017, -0.78986972877791872,

/\* layer 4 \*/

0.69411327921155475

};

static double NNCode01Weights[] =

{

/\* layer 1 \*/

0.45042301341189306, 2.1217686262666771, -1.0068156383760065,

-1.1413260174080213, -1.0991150710499151, 0.13897448897109849,

-0.29607430226359727, 1.0578000077266938, 0.015375000112306594,

-1.2422364426805548, -0.74956858709525265, -0.8553271347659136,

-0.85607274995188631, 0.29497405886945072, 0.77736824062384691,

-0.20929848326342507, -0.22853697200549497, 0.74891389540388054,

/\* layer 2 \*/

1.7021771309298774, -0.53360324411020343, -0.31156922875290471, -0.091379036912736369,

-1.4989653978827395, 0.72340976268314061,

1.4437813548042771, -0.89535616028769582, 1.1216157458629343, 1.3274205702071624,

-1.1352529561171858, 0.14210804706797561,

0.41425714287406146, 0.431330713205753, -0.43841195501747676, -0.44254618991439187,

0.42331552646000659, 0.94711850180826818,

-0.69777169749768897, -0.31411131517812213, 0.37626203368294159, 0.72220124168593791,

1.4440010516722501, 0.42991920561824032,

-1.8873198409426812, 1.395436121708318, -0.11449816108829589, 1.0091510500308545,

1.4996656748816071, -0.41794997206613183,

0.78354761133001638, -2.5494762135506481, 0.40882260130178899, 1.5910666675546739,

0.80999043614787636, -0.29279559333382182,

/\* layer 3 \*/

-0.25577187297085369, -0.054086526409348461, -3.6366497275630278, -1.1475005269223759,

-1.1116567649541851, -2.1706932802718115,

0.27443029693461668, -0.48560917012506649, -0.56188985673386882, -0.14286841030395986,

-1.0178217629357849, 1.195356783569268,

0.29726106579120304, -0.49397451835068895, -1.0861619028553853, -0.70294227300573964,

0.40998926187701779, -0.57592736511868514,

-1.1986750623331395, -1.4375085686817586, -0.55873088348972355, -0.87617424627912732,

-0.35352994548403482, -0.33820170592285714,

-0.086633752963811456, 1.1553463079379975, 0.56447798034575358, -0.26676926256425054,

0.36756489192674663, 1.3357814037735765,

1.1843768719771028, 1.1108495556704983, -0.90088831737647312, -0.22970074807497515,

-0.41786589316512507, 0.42643897459447383,

/\* layer 4 \*/

0.99063067386747738, -1.8396681665753518, -0.98949246393556289, -1.2881741957923629,

0.24298754999365188, -0.43613008677115006

};

static double NNCode01Acts[44];

/\* ---------------------------------------------------------- \*/

/\*

NNCode01Run - run neural network NNCode01

Input and Output variables.

Variable names are listed below in order, together with each

variable's offset in the data set at the time code was

generated (if the variable is then available).

For nominal variables, the numeric code - class name

conversion is shown indented below the variable name.

To provide nominal inputs, use the corresponding numeric code.

Input variables (Offset):

X (0)

Y (1)

Z (2)

Âûõîä:

Class1 (3)

1=0

2=1

\*/

/\* ---------------------------------------------------------- \*/

void NNCode01Run( double inputs[], double outputs[], int outputType )

{

int i, j, k, u;

double \*w = NNCode01Weights, \*t = NNCode01Thresholds;

/\* Process inputs - apply pre-processing to each input in turn,

\* storing results in the neuron activations array.

\*/

/\* Input 0: standard numeric pre-processing: linear shift and scale. \*/

if ( inputs[0] == -9999 )

NNCode01Acts[0] = 0.52964743589743613;

else

NNCode01Acts[0] = inputs[0] \* 0.020833333333333332 + -2.1666666666666665;

/\* Input 1: copy input into activations. \*/

if ( inputs[1] == -9999 )

NNCode01Acts[1] = 129;

else

NNCode01Acts[1] = inputs[1];

/\* Input 2: standard numeric pre-processing: linear shift and scale. \*/

if ( inputs[2] == -9999 )

NNCode01Acts[2] = 0.76946176726772453;

else

NNCode01Acts[2] = inputs[2] \* 0.0068086167947741952 + 0.012750564757741584;

/\*

\* Process layer 1.

\*/

/\* For each unit in turn \*/

for ( u=0; u < 6; ++u )

{

/\*

\* First, calculate post-synaptic potentials, storing

\* these in the NNCode01Acts array.

\*/

/\* Initialise hidden unit activation to zero \*/

NNCode01Acts[3+u] = 0.0;

/\* Accumulate weighted sum from inputs \*/

for ( i=0; i < 3; ++i )

NNCode01Acts[3+u] += \*w++ \* NNCode01Acts[0+i];

/\* Subtract threshold \*/

NNCode01Acts[3+u] -= \*t++;

/\* Now apply the hyperbolic activation function, ( e^x - e^-x ) / ( e^x + e^-x ).

\* Deal with overflow and underflow

\*/

if ( NNCode01Acts[3+u] > 100.0 )

NNCode01Acts[3+u] = 1.0;

else if ( NNCode01Acts[3+u] < -100.0 )

NNCode01Acts[3+u] = -1.0;

else

{

double e1 = exp( NNCode01Acts[3+u] ), e2 = exp( -NNCode01Acts[3+u] );

NNCode01Acts[3+u] = ( e1 - e2 ) / ( e1 + e2 );

}

}

/\*

\* Process layer 2.

\*/

/\* For each unit in turn \*/

for ( u=0; u < 6; ++u )

{

/\*

\* First, calculate post-synaptic potentials, storing

\* these in the NNCode01Acts array.

\*/

/\* Initialise hidden unit activation to zero \*/

NNCode01Acts[9+u] = 0.0;

/\* Accumulate weighted sum from inputs \*/

for ( i=0; i < 6; ++i )

NNCode01Acts[9+u] += \*w++ \* NNCode01Acts[3+i];

/\* Subtract threshold \*/

NNCode01Acts[9+u] -= \*t++;

/\* Now apply the hyperbolic activation function, ( e^x - e^-x ) / ( e^x + e^-x ).

\* Deal with overflow and underflow

\*/

if ( NNCode01Acts[9+u] > 100.0 )

NNCode01Acts[9+u] = 1.0;

else if ( NNCode01Acts[9+u] < -100.0 )

NNCode01Acts[9+u] = -1.0;

else

{

double e1 = exp( NNCode01Acts[9+u] ), e2 = exp( -NNCode01Acts[9+u] );

NNCode01Acts[9+u] = ( e1 - e2 ) / ( e1 + e2 );

}

}

/\*

\* Process layer 3.

\*/

/\* For each unit in turn \*/

for ( u=0; u < 6; ++u )

{

/\*

\* First, calculate post-synaptic potentials, storing

\* these in the NNCode01Acts array.

\*/

/\* Initialise hidden unit activation to zero \*/

NNCode01Acts[15+u] = 0.0;

/\* Accumulate weighted sum from inputs \*/

for ( i=0; i < 6; ++i )

NNCode01Acts[15+u] += \*w++ \* NNCode01Acts[9+i];

/\* Subtract threshold \*/

NNCode01Acts[15+u] -= \*t++;

/\* Now apply the hyperbolic activation function, ( e^x - e^-x ) / ( e^x + e^-x ).

\* Deal with overflow and underflow

\*/

if ( NNCode01Acts[15+u] > 100.0 )

NNCode01Acts[15+u] = 1.0;

else if ( NNCode01Acts[15+u] < -100.0 )

NNCode01Acts[15+u] = -1.0;

else

{

double e1 = exp( NNCode01Acts[15+u] ), e2 = exp( -NNCode01Acts[15+u] );

NNCode01Acts[15+u] = ( e1 - e2 ) / ( e1 + e2 );

}

}

/\*

\* Process layer 4.

\*/

/\* For each unit in turn \*/

for ( u=0; u < 1; ++u )

{

/\*

\* First, calculate post-synaptic potentials, storing

\* these in the NNCode01Acts array.

\*/

/\* Initialise hidden unit activation to zero \*/

NNCode01Acts[21+u] = 0.0;

/\* Accumulate weighted sum from inputs \*/

for ( i=0; i < 6; ++i )

NNCode01Acts[21+u] += \*w++ \* NNCode01Acts[15+i];

/\* Subtract threshold \*/

NNCode01Acts[21+u] -= \*t++;

/\* Now apply the logistic activation function, 1 / ( 1 + e^-x ).

\* Deal with overflow and underflow

\*/

if ( NNCode01Acts[21+u] > 100.0 )

NNCode01Acts[21+u] = 1.0;

else if ( NNCode01Acts[21+u] < -100.0 )

NNCode01Acts[21+u] = 0.0;

else

NNCode01Acts[21+u] = 1.0 / ( 1.0 + exp( - NNCode01Acts[21+u] ) );

}

/\* Type of output required - selected by outputType parameter \*/

switch ( outputType )

{

/\* The usual type is to generate the output variables \*/

case 0:

/\* Post-process output 0, two-state nominal output \*/

if ( NNCode01Acts[21] >= 0.5 )

outputs[0] = 2.0;

else

outputs[0] = 1.0;

break;

/\* type 1 is activation of output neurons \*/

case 1:

for ( i=0; i < 1; ++i )

outputs[i] = NNCode01Acts[21+i];

break;

/\* type 2 is codebook vector of winning node (lowest actn) 1st hidden layer \*/

case 2:

{

int winner=0;

for ( i=1; i < 6; ++i )

if ( NNCode01Acts[3+i] < NNCode01Acts[3+winner] )

winner=i;

for ( i=0; i < 3; ++i )

outputs[i] = NNCode01Weights[3\*winner+i];

}

break;

/\* type 3 indicates winning node (lowest actn) in 1st hidden layer \*/

case 3:

{

int winner=0;

for ( i=1; i < 6; ++i )

if ( NNCode01Acts[3+i] < NNCode01Acts[3+winner] )

winner=i;

outputs[0] = winner;

}

break;

}

}

/\* ---------------------------------------------------------- \*/

/\*

NNCode01RunPadded - network NNCode01

inputs - the input variables, in the same number and order

as in the data set at the time the code was generated.

This alternative routine is useful if you want a consistent

interface for your generated routines, so that the number

and order of variables is the same for all of them.

Variables (ones used as inputs marked thus \*):

0) X \*

1) Y \*

2) Z \*

Âûõîä:

Class1 (3)

1=0

2=1

\*/

/\* ---------------------------------------------------------- \*/

void NNCode01RunPadded( double inputs[], double outputs[], int outputType )

{

double in[3];

/\* Copy inputs \*/

in[0]=inputs[0];

in[1]=inputs[1];

in[2]=inputs[2];

/\* Run the network \*/

NNCode01Run( in, outputs, outputType );

}

/\*

Test harness. Compile including this main() procedure, as

a windows console program or a DOS program, to interactively

test that the software functions as expected.

\*/

int main(void)

{

int i, outputType=0, noOutputs=1;

double inputs[3], outputs[3];

printf( "\n\nNNCode test harness program. Enter inputs below\n" );

printf( "Nominal variables should be numbered starting at 1 (0 for missing)\n" );

printf( "(e.g. if an input is Gender={male,female}, enter 1 for male, 2 for female)\n" );

/\* Infinite (user-breakable) loop for repeated tests \*/

start\_of\_loop:

while ( 1 )

{

/\* Get the input pattern \*/

for ( i=0; i < 3; ++i )

{

printf( "Enter value for input %d: ", i+1 );

scanf( "%lg", & inputs[i] );

/\* Check for sub-menu \*/

if ( inputs[i] == MENUCODE )

{

printf( "Control menu. Select output style, or exit:\n" );

printf( "0. Normal output style (output variable)\n" );

printf( "1. Output layer activations\n" );

printf( "2. Codebook vector (usual only for Kohonen networks\n" );

printf( "3. Winning hidden neuron (ditto only Kohonen)\n" );

printf( "4. Exit program\n" );

printf( "> " );

scanf( "%d", & outputType );

if ( outputType < 0 || outputType > 3 )

{

printf( "\nBye.\n" );

return 0;

}

else

{

/\* Determine how many outputs there are to display (depends on

\* output type)

\*/

switch ( outputType )

{

case 0:

noOutputs = 1;

break;

case 1:

noOutputs = 1;

break;

case 2:

noOutputs = 3;

break;

case 3:

noOutputs = 1;

break;

}

goto start\_of\_loop;

}

}

}

/\* Run the neural network \*/

NNCode01Run( inputs, outputs, outputType );

/\* Display the output of the neural network \*/

printf( "\n\nOutput of neural network:\n" );

for ( i=0; i < noOutputs; ++i )

{

printf( "Output %d: ", i+1 );

printf( "%g\n", outputs[i] );

}

printf( "\nEnter next input pattern (for control menu inc. exit, enter %d for any input):\n", MENUCODE );

}

return 0;

}