ЗАТВЕРДЖЕНО

Наказ Міністерства освіти і науки, України

29.03.2012 № 384

**Форма № Н-9.02**

Національний університет «Полтавська політехніка ім. Ю.Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки

(повна назва інституту)

Кафедра комп’ютерних та інформаційних технологій і систем 

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

бакалавра 

(освітній рівень)

на тему

Розробка апаратно-програмних засобів нейромережевої моделі оперативної діагностики стану здоров’я

Виконав: студент 4 курсу, групи 402-ТК

спеціальності

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

(шифр і назва спеціальності)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Швець Д.Р.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

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

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМ. Ю.КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

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

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

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

**на тему**

**«Розробка апаратно-програмних засобів нейромережевої моделі оперативної діагностики стану здоров’я»**

##### **Студента групи 402-ТК Швець Д.Р.**

Керівник роботи

кандидат технічних наук,

доцент Альошин С.П.

Завідувач кафедри

кандидат технічних наук,

доцент Головко Г.В.

Полтава – 2021**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 61 сторінок, 35 рисунок, 6 джерел, 1 додаток.

**Об’єкт дослідження:** Інформаційне забезпечення діагностики стану здоров’я

**Предмет дослідження:** нейромережева модель стану здоров’я

**Мета роботи**: дослідження нейромережевих технологій визначення стану самопочуття організму у задачі створення нейромережевого забезпечення стану здоров’я.

**Методи**: математичний аналіз, нейромережевий аналіз, класифікація, регресія, кластеризація.

**Ключові слова**: нейромережеве забезпечення, інтелектуальний класифікатор, простір ознак, нейромережеві технології, визначення стану здоров’я.

**ABSTRACT**

Bachelor's qualification work**:** 61 pages., 35 pictures., 6 sources, 1 addition.

**Object of research:** Information support of health diagnostics

**Subject of research:** neural network model of health status

**Purpose:** research of neural network technologies to determine the state of health of the organism in the task of creating neural network health support.

**Methods:** mathematical analysis, neural network analysis, classification, regression, clustering.

**Keywords:** neural network support, intelligent classifier, feature space, neural network technologies, health determination.

**ЗМІСТ**

[РОЗДІЛ 1 8](#_Toc74318054)

[ПОСТАНОВКА ЗАДАЧІ 8](#_Toc74318055)

[РОЗДІЛ 2 10](#_Toc74318056)

[ОСНОВНІ ПОНЯТТЯ ПРО ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ 10](#_Toc74318057)

[2.1 Інтелектуальний аналіз даних 10](#_Toc74318058)

[2.2 Основні поняття про експертні системи. 18](#_Toc74318059)

[2.2.1 Моделювання кінцевих етапів розробки ЕС. 19](#_Toc74318060)

[РОЗДІЛ 3 26](#_Toc74318061)

[3.1. Моделювання етапів розробки ЕС для заданої предметної області. 26](#_Toc74318062)

[3.2 Етап розробки 26](#_Toc74318063)

[3.2.1 Етапи створення нової мережі: 27](#_Toc74318064)

[3.2.2 Робота з таблицею даних 28](#_Toc74318065)

[3.3 Навчання нейронної мережі 30](#_Toc74318066)

[3.4 Режим інтерактивного навчання 33](#_Toc74318067)

[3.5 Методика рішення задачі 36](#_Toc74318068)

[3.5.1 Вибір та аналіз нейроархітектури: 38](#_Toc74318069)

[3.5.2 Аналіз результатів навченої мережі: 39](#_Toc74318070)

[РОЗДІЛ 4 44](#_Toc74318071)

[4.1 РОЗВЯЗАННЯ ЗАДАЧІ 44](#_Toc74318072)

[ВИСНОВКИ 55](#_Toc74318073)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 56](#_Toc74318074)

[ДОДАТОК А 57](#_Toc74318075)

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

**ШІ –** штучний інтелект, здатність інженерної системи (англ. engineered system) здобувати, обробляти та застосовувати знання та вміння.

**ШНМ (ANN)** **–** це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин.

**ЕОМ –** електронна обчислювальна машина, загальна назва для обчислювальних машин, що є електронними.

**Data Mining –** процес напівавтоматичного аналізу великих баз даних з метою пошуку корисних фактів.

**ЕС –** Експертна система, це методологія адаптації алгоритму успішних рішень однієї сфери науково-практичної діяльності в іншу.

ВСТУП

Інформаційні системи із штучним інтелектом це кібернетичні системи, але вони тільки частина кібернетики. Кібернетика, за Норбертом Вінером, — це наука про управління і зв’язок у живому організмі і машині[1]. Проблеми створення ШІ пов’язуються з комп’ютерними системами «інтелектуальною» поведінкою.

Поки що немає чіткого визначення поняття інтелекту, але частіше за все, під цим визначенням розуміють здатність машини правильно відноситись до нової ситуації, тим більше, коли мова йдеться про штучний інтелект. Системами з штучним інтелектом можна назвати системи обчислювальних і логічних машин, що виконують властиві людині інтелектуальні функції, пов’язані зі сприйманням та обробкою знань, міркуванням і відповідним спілкуванням. Для відповіді на питання: – «Які машини можна назвати Інтелектуальними», Алан Тюрінг запропонував експериментатору спілкуватися через третю особу з невидимими йому людиною чи машиною. Інтелектуальною можна вважати ту машину, яка в процесі такого спілкування не відрізняється від людини. Але в наш час, інтелектуальними також називають людино-машинні системи, що компетентно використовують знання людини та логічні можливості машини.

Академік Г.С. Поспєлов наголошує – «Процес створення ШІ йде двома напрямками. Перший, біонічний — спроба змоделювати діяльність мозку в перспективі відтворити на ЕОМ штучний розум. Другий — прагматичний, дозволяє засобами ЕОМ відтворити процеси розумової діяльності людини.

Дослідження ШІ прямує в напрямку відтворення на ЕОМ спеціальних функцій творчих процесів; зовнішньої інтелектуалізації ЕОМ поліпшенням діалогового інтерфейсу; внутрішньої інтелектуалізації ЕОМ з використанням нової архітектури на принципах ШІ; створення інтелектуальних робото-технічних пристроїв з можливістю аналізу ситуації і автономного вироблення рішень.

## РОЗДІЛ 1

## ПОСТАНОВКА ЗАДАЧІ

Зв'язок індикаторів цільової функції (станів об'єкта) з простором описуючих їх ознак, відобразимо так:

 (1.1)

де  **,

де (X → ... .X) - 1, 2, ... - простір ознака,

F- Способи перетворення ознака та визначення класів,

L - розмірність ознака в просторах 1, 2, ..., n,

k - число досліджуваних класів (розмірність класів).

Масив вибірок інформативних ознак станів досліджуваного об'єкта і класів цих станів представимо в загальному вигляді:

 (1.2)

спільно з відповідним масивом класів  (класи станів) використовується для навчання в задачі розпізнавання актуальних станів об'єкта дослідження. Цю задачу можна сформулювати більш предметно і конкретно: оптимізувати функціонал процесу розпізнавання поточного стану об'єкта:

   (1.3)

   де  - правило віднесення об'єкта  до відповідного класу;

 - безліч об'єктів в класі.

Рішення (1) досягається реалізацією перетворення

→ , згідно вирішального правила: якщо

= (1.4)

то 

Інструментальна процедура розпізнавання можлива і доцільна з застосування стандартних нейроемуляторів існуючих пакетів технічного аналізу даних. При цьому порівняння ознак в класах здійснюється в деякому метричному форматі, наприклад, у форматі евклідової метрики:

 (1.5)

де представлені ознаки в класах (стани об'єкта) *p,g*, при аналізі *k,l* - об'єктів - *j* типу).

*j* - тип ознак являє собою елемент дискретного простору ознак.

Таким чином, маємо формальну оптимізаційну задачу, реалізація якої вимагає програмно-інструментального обґрунтування і адекватних моделей для отримання практичних результатів.

## РОЗДІЛ 2

## ОСНОВНІ ПОНЯТТЯ ПРО ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

## 2.1 Інтелектуальний аналіз даних

Процес дослідження даних визначається стандартами, які містять покрокові рекомендації, задачі та цілі для всіх етапів процесу аналізу даних. На етапі «Підготовки даних» створюється і вивчається початковий набір даних, для визначення, придатності для наступної обробки. Якщо кількість даних має невеликий об’єм, буде потрібно провести збір нових даних більш строгих критеріїв[14].

Етап «Передoобробки даних» містить попередню обробку необроблених даних, для побудови аналітичної моделі, методами машинного навчання здатні розробити аналітичну модель, відображаючи структурний опис інформації, яка міститься в учбових даних. Етап «Побудова моделі» –– створюється внутрішня модель даних для перевірки. Етапи «Підготовка даних» і «Побудова моделі» виконуються паралельно. Переважно необхідно їх повторювати, через те що, результати отримані в ході відтворення, забезпечують краще розуміння вимог до методів попередньої обробки. Вагомим етапом є «Тестування моделі», інакше кажучи, перевірки побудованої аналітичної моделі. За допомогою тестування робимо висновок, чи може модель, побудована за допомогою навчальних даних, точно класифікувати тестові дані. Існує багато методів тестування побудованих аналітичних моделей. Коли етап тестування відображає, що модель є незадовільною, то потрібно переглядати весь проект аналізу даних. Якщо точність моделі достатньо висока, тоді виконується етап «Розгортання моделі», для цього модель передається інженерам з програмного забезпечення. Мета інтелектуального аналізу даних –– побудова аналітичних моделей, оптимізованих для вирішення прикладних задач. Аналітичні моделі будуються для навчальних даних з конкретної проблемної області [2].

Коли тільки модель побудувалась, вона перевіряється на тестових даних. Після підтвердження точності побудованої аналітичної моделі, модель можлива вдаватися до нових некласифікованих даних. Відповідно до стилю пошуку шаблонів в освітніх даних, методи видобутку даних поділяються на такі категорії: - навчання з викладачем – модель базується на попередньо класифікованих освітніх даних і описує закономірності між значеннями атрибутів та значення атрибутів класу; – навчання без учителя: в освітніх даних немає аспекту класу, шукаються закономірності між значеннями всіх аспектів, які розглядаються аспектами ознак; - часткове навчання з викладачем – обсяг класифікованих даних навчання набагато менше обсягу некласифікованих даних, тому спочатку для класифікованих даних вирішується проблема навчання з викладачем і будується аналітична модель, потім проблема навчання без вчитель вирішень; - Посилене навчання – з появою нових навчальних даних попередня модель не перебудовується, але успадковує знайдені раніше шаблони, які показали правильні результати класифікації на тестових наборах даних.

Популярність ***Data Mining*** сьогодні можна порівняти з популярністю цієї області півстоліття тому, на зорі комп’ютерної ери. У той час, однак, цей термін не був відомий, але багато говорили про такі поняття як: Штучний Інтелект та Нейронні Мережі. Однак, за невеликими винятками, практичну реалізацію теорії довелося відкласти, поки апаратна та програмна інфраструктура не розвинеться до сучасного рівня. І сьогодні, після 50-річного циклу розробки, ми знову звертаємось до вирішення проблем аналізу, маючи вже потужні комп’ютерні системи та системи управління базами даних, розвинене операційне та мовне середовище.

Структура методів ***Data Mining***представлена на рисунку 2.1

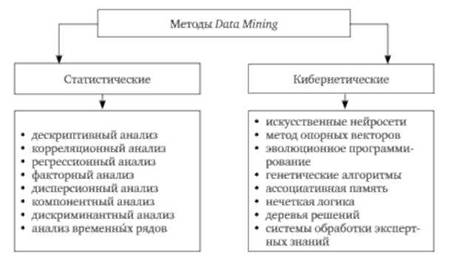


Рисунок 2.1 – Методи інтелектуального аналізу даних

Методи ***Data Mining*** діляться на дві групи:

- Контрольоване навчання;

- Навчання без нагляду.

У першому випадку завдання аналізу даних, наприклад класифікація, виконується в кілька етапів. Це один із методів машинного навчання, під час якого перевірена система змушена вчитися на прикладах. Між входами та опорними виходами може існувати певна залежність, але це невідомо. Відомий лише кінцевий набір прецедентів, який називається навчальним зразком. На основі цих даних необхідно відновити залежність, тобто побудувати алгоритм, здатний видавати будь-який об'єкт досить точну відповідь. Для вимірювання точності відповідей, а також при навчанні прикладів можна ввести якісний функціонал. Спочатку за допомогою деякого алгоритму Data Mining будується модель аналізованих даних - класифікатор. Тоді класифікатор підлягає "засвоєнню" [6]. Іншими словами, перевіряється якість його роботи, і якщо вона є незадовільною, відбувається "додаткове навчання" класифікатора. Це триває до тих пір, поки ми не досягнемо необхідного рівня якості або не переконаємось, що обраний алгоритм не працює належним чином з даними або самі дані не мають структури, яку можна виявити.

Машинне навчання без нагляду - один із способів машинного навчання. З його допомогою перевірена система спонтанно вчиться виконувати завдання без втручання експериментатора. Як правило, він підходить лише для задач, в яких відомі описи сукупності об'єктів (навчальної вибірки), і необхідно виявити внутрішні взаємозв'язки, залежності, закономірності, що існують між об'єктами. Наприклад, схеми покупок, здійснених покупцями великого магазину. Очевидно, що якщо ці закономірності існують, то модель повинна їх представити та недоречно говорити про своє навчання. Звідси і назва «навчання без вчителя».

Навчання без вчителя часто протиставляється навчанню з учителем, коли для кожного навчального об’єкта змушують дати «правильну відповідь», потрібно знайти взаємозв’язок між стимулами та системними реакціями.

**2.1.1 Прикладнi моделi прогнозування.** Інша класифікація завдань машинного навчання виникає при розгляді бажаного результату роботи системи за допомогою машинного навчання:

**У класифікації**. Вхідні дані поділяються на два або більше класів, і система студентів повинна генерувати модель, яка призначає безпрецедентні вхідні дані одному або декільком з цих класів. Зазвичай це намагаються вирішити контрольовано. Прикладом класифікації є спам-фільтри, в які вводяться електронні листи (або щось інше), а класи - "спам" та "не спам".

**У регресії**. Кероване завдання, результати є безперервними, не дискретними.

**У кластеризації.** Набір вхідних даних слід розділити на групи. На відміну від класифікації, групи не відомі заздалегідь, що зазвичай робить це завданням для спонтанного навчання.

**Еволюційне навчання.** (англ. developmental learning), призначений для навчання роботів, генерує власні послідовності навчальних ситуацій (також званих навчальними планами) для накопичення репертуару нових навичок за допомогою автономного самодослідження та соціальної взаємодії з вчителями-людьми та використання провідних механізмів, таких як активне навчання, дозрівання, рухова синергія та імітація.

**Навчання дерев рішень.** Навчання дерев рішень (англ. decision tree learning) використовує як передбачувальну модель дерево рішень, яке відображує спостереження про предмет на висновки про цільове значення предмету [4].

**Навчання асоціативних правил.** Навчання асоціативних правил (англ. association rule learning) є методом відкривання цікавих зв'язків між величинами у великих базах даних [4].

**Штучні нейронні мережі.** Алгоритм навчання штучної нейронної мережі (ANN), відомий як "нейронна мережа" (NN), - це алгоритм дистанційного навчання, натхненний біологічними нейронними мережами. Структурні обчислення структуровані з точки зору взаємодії споріднених груп штучних нейронів, які обробляють інформацію, використовуючи коннективістський підхід до обчислень. Сучасні нейронні мережі - це нелінійні засоби статистичного моделювання даних. Їх слід використовувати для моделювання складних взаємозв’язків між входами та результатами, для пошуку закономірностей даних або для виявлення статистичної структури в неформальній спільноті розподілу безпеки спостережуваних величин. [4].

**Глибинне навчання.** Падіння цін на обладнання та розвиток графічних процесорів для особистого користування протягом останніх кількох років сприяли розробці концепції глибокого навчання, яке створюється з декількох прихованих шарів штучної нейронної мережі. Цей підхід спрямований на спрощення моделювання, коли мозок людини поширює світло та звук для зору та слуху. Деякі з успішних програм глибокого навчання - це комп’ютерний зір та розпізнавання мови [4].

**Індуктивне логічне програмування.** Індуктивне логічне програмування (ILP) - це підхід до вивчення правил логічного програмування як універсального подання вхідних складів, зворотного розширення та гіпотези. Завдяки кодуванню відомих базових знань та набору складів, представлених як логічна база даних, система ILP виведе гіпотетичну логічну програму, яка має всі позитивні та негативні результати з негативними компонентами. Суміжною областю є індуктивне програмування, яке вважається гіпотезою кожного, хто бачить мову програмування (а не лише логічне програмування), а також функціональні програми. [4].

**Метод опорних векторів.** Метод опорних векторних машин (SVM) - це сукупність методів навчання, пов’язаних з учителями, що використовуються для класифікації та регресії. Набір навчальних прикладів, кожен з яких позначений як приналежний до однієї з двох категорій, алгоритм навчальних опорних векторів будує модель, яка передбачає, чи підпадає новий приклад до тієї чи іншої категорії. [4].

**Кластерування.** Кластерний аналіз - це розподіл набору спостережень на підмножини (так звані кластери), так що спостереження в межах одного кластеру схожі на дію раніше створеного кар'єру або артерій, тоді як спостереження, взяті з різних скупчень, неоднакові. Різні методи кластеризації здійснюють різну підготовку щодо структури даних, часто визначають ефективні міри подібності та оцінюють, наприклад, внутрішню компактність (подібні члени одного кластера) та чіткість між різними кластерами. Інші методи засновані на оцінці щільності та графіках зв’язку. Кластеризація - це метод навчання без вчителя, а також загальноприйнятий метод статистичного аналізу даних [4].

**Баєсові мережі.** Баєсова мережа, мережа переконань або спрямована ациклічна графічна модель - це імовірнісна модель графіка, яка представляє набір випадкових величин та їх умовні незалежності через спрямований ациклічний графік. Наприклад, баєсова мережа може представляти ймовірнісний зв’язок між хворобою та симптомами. Маючи симптоми, за допомогою такої мережі можна розрахувати ймовірність різних захворювань. Існують ефективні алгоритми для виведення та навчання [4].

**Навчання з підкріпленням.** Навчання з підкріплення стосується того, як агент повинен діяти в навколишньому середовищі таким чином, щоб максимізувати якесь поняття довгострокової винагороди. Посилені алгоритми навчання намагаються знайти політику, яка відображає держави світу щодо дій, які агент повинен виконувати в цих штатах. Посилене навчання відрізняється від навчання вчителя тим, що пари правильних входів / результатів йому ніколи не представляються, і не зовсім оптимальні дії ніколи чітко не коригуються. [4].

**Навчання представлень.** Деякі алгоритми навчання, переважно алгоритми навчання без вчителів, мають на меті відкрити найкращі уявлення про вхідні дані, що надаються під час навчання. Класичні приклади включають метод основних компонентів та кластерний аналіз. Представницькі алгоритми навчання часто намагаються зберігати інформацію у своїх вхідних даних, але перетворюють її, щоб зробити її зручною, часто як етап попередньої обробки перед виконанням класифікації або прогнозування, дозволяючи відновити вхідні дані з невідомого розподілу. генерує дані, хоча це не обов’язково є точним для конфігурацій, які навряд чи підпадають під такий розподіл [6].

**Алгоритми навчання многовидів** намагаються робити це за обмеження, щоби навчені представлення мали низьку розмірність. Розріджені алгоритми кодування намагаються зробити це, обмежуючи вивчені подання розрідженими (мають багато нулів). Алгоритми вивчення полі-лінійного підпростору, спрямовані на вивчення низьковимірних подань безпосередньо з тензорних подань багатовимірних даних без перетворення їх у (багатовимірні) вектори [6].

**Алгоритми глибинного навчання** відкривають кілька рівнів представлення про ієрархію ознак, в яких високорівневі, більш абстрактні ознаки визначаються з точки зору ознак нижчого рівня (або генерують їх). Існує припущення, що розумна машина - це та, яка засвоює ідею, яка розкриває фактори, що лежать в основі змін, що описують спостережувані дані. [6].

**Навчання подібностей та мір.** У цій задачі навчальній машині подано пари прикладів, які вважаються подібними, і пари менш подібних об’єктів. Їй потрібно вивчити функцію подібності (або функцію вимірювання відстані), яка може передбачити, чи схожі нові об’єкти. Це іноді використовується в реферальних системах [6].

## 2.2 Основні поняття про експертні системи

Експертна система (ЕС) - це система підтримки прийняття рішень, яка містить знання з конкретної вузької предметної області, а також може запропонувати користувачеві рішення проблем у цій галузі та обґрунтувати їх. Експертна система складається з бази знань, механізму логічного висновку та підсистеми обґрунтування[12].

Експертна система акумулює професійні знання менеджерів та спеціалістів, використовуючи їх для формування бази знань, що містить набір взаємопов’язаних правил. Приймаючи рішення, стає можливим проаналізувати наслідки різних рішень у формі запитань "що буде, якщо ...", не витрачаючи час на трудомісткий процес програмування.

Створення експертних систем - це спроба значно розширити сферу обчислювальної техніки та значно збільшити її можливості як допоміжного засобу для людини в його інтелектуальній праці[12].

До появи комп'ютерів з експертною системою були створені принципи алгоритмічної методології. Для того, щоб такі обчислювальні системи працювали успішно, повинен бути дотриманий значний набір передумов. Перш за все, для кожного розв'язуваного завдання потрібно знайти або створити алгоритм. Потім цей алгоритм повинен бути перетворений у звіт програми, що реалізує майбутні розрахунки. Крім того, потрібно переконатися, що всі думки забезпечені повною кількістю достовірної джерельної інформації [4].

Таке використання комп'ютерного обладнання пов'язане зі значними труднощами. По-перше, із збільшенням складності завдань швидко зростає складність та вартість програмування робочого комп’ютера, що вже стає сильним гальмом для подальшого використання комп’ютерних технологій. По-друге, багато практичних завдань, що виконують процес у процесі людської діяльності, не забезпечуються доступними загальнодоступними джерелами даних, якими люди зазвичай користуються в умовах більшої чи меншої інформаційної невизначеності[15].

В результаті багато важливих завдань людина не може перекласти на комп’ютері, який працює за принципами алгоритмічної методології, що значно скорочує сферу практичного застосування комп’ютерних технологій. Разом з цим, людина успішно справляється з подібними задачами завдяки своєму вмінню:

– працювати з задачами, не здійснюючи їх повну формалізацію;

– знаходити і використовувати для розв'язання задач різноманітні, найнеочікуваніші джерела інформації;

– поєднувати суперечливі відомості, надаючи їм потрібну інтерпретацію

– цілеспрямоване вивчення або перевірка об'єкту дії для одержання додаткової інформації;

– відтерміновувати рішення до накопичення потрібних даних;

– продукувати неоднозначні рішення і успішно керуватися ними;

– створювати і накопичувати знання, які уможливлюють діяти в умовах значної інформаційної невизначеності.

Спроба наділити комп'ютери переліченими цінними характеристиками призвела до створення експертних систем [9].

2.2.1 Моделювання кінцевих етапів розробки ЕС. Етап ідентифікації включає розуміння завдань, що стоять перед ЕС, та встановлення вимог до нього. Результатом цього етапу є відповідь на питання, що потрібно зробити і які ресурси потрібно використовувати (визначення завдань, визначення учасників процесу проектування та їх ролі, визначення ресурсів та цілей) [9].

Ідентифікація завдання полягає у складанні неформального (словесного) опису, в якому зазначається: загальна характеристика завдання; підзадачі, визначені в цьому завданні; ключові поняття (об'єкти), їх вхідні (вихідні) дані; можливий тип рішення, а також знання, пов'язані з проблемою.

Початковий неформальний опис завдання експертом використовується інженером знань для уточнення термінів та ключових понять. Експерт виправляє опис проблеми, пояснює, як її вирішити і які міркування лежать в основі конкретного рішення. Після кількох циклів вдосконалення опису фахівець та інженер знань отримують остаточний неформальний опис завдання [9].

Типовими ресурсами для розвитку ЕС є джерела знань, час розробки, обчислювальні ресурси та фінансування. Для експерта джерелами знань є його попередній досвід вирішення проблем, книги, відомі приклади вирішення проблем, а для інженера знань - досвід вирішення подібних проблем, методи подання та маніпулювання знаннями, програмне забезпечення.

При встановленні цілей важливо розрізняти цілі, заради яких був створений ЕС, та завдання, які він повинен вирішувати. Прикладами може слугувати:

формалізація неформальних знань експертів, поліпшення якості прийнятих експертом рішень, автоматизація рутинних аспектів роботи експерта (користувача), відтворення знань експерта [10].

**Етап концептуалізації**. На цьому етапі проводиться змістовний аналіз проблемної області, визначаються використовувані поняття та їх взаємозв'язки, визначаються методи вирішення проблем. Цей етап закінчується створенням моделі предметної області (СО), що включає основні поняття та взаємозв'язки. На етапі концептуалізації визначаються такі особливості завдання: типи доступних даних, вихідні та вхідні дані, підзадачі загальної проблеми, використовувані стратегії та гіпотези, типи взаємозв'язків між програмними об'єктами, типи використовуваних відносин (ієрархія, причина - наслідок, частина - ціле та ін.), процеси, що використовуються при прийнятті рішення, склад знань, що використовуються при вирішенні проблеми, типи обмежень, що накладаються на процеси, що використовуються у вирішенні, склад знань, використаних для обґрунтування рішення [2].

**Етап формалізації*.*** Зараз усі ключові поняття та відносини виражаються формальною мовою, яка або вибирається з числа існуючих, або відтворюється заново. Іншими словами, на цьому етапі визначається склад засобів і методів подання декларативних та процедурних знань, проводиться ця презентація і, нарешті, формується опис вирішення проблеми ЄС формальною мовою, запропонованою інженером знань. Результатом етапу формалізації є опис того, як розглянуту проблему можна представити у вибраному або розвиненому формалізмі. А також зазначення способів подання знань (кадри, сценарії, семантичні мережі тощо) та визначення способів маніпулювання цими знаннями (логічний висновок, аналітична модель, статистична модель тощо) та інтерпретація знань [2].

**Етап виконання**. Метою цього етапу є створення одного або декількох прототипів ЄС, вирішення необхідних завдань. Потім на цьому етапі для результатів випробувань та дослідницьких робіт створюється кінцевий продукт, доданий для промислового використання. Розробка прототипу проявляється у програмуванні його компонентів або їх підборі відомими інструментами та заповненні бази знань [2].

Головне при створенні прототипу полягає в тому, що цей прототип забезпечує перевірку відповідності ідей, методів та можливостей надання важливих важливих справ. Створення першого прототипу, очевидно, підтверджує, що він формує методологію рішень та шляхи представлення додаткових для успішного вирішення принаймні ряду проблем з відповідною предметною областю, і демонструє тенденцію до отримання якісних та ефективних рішень для всіх, хто шукає предметну область в помірних кількостях [6]. При розробці ЕС, крім перерахованих завдань, вирішуються такі:

1. Аналіз функціонування системи при значному розширенні бази знань;
2. Дослідження можливостей системи в рішенні більш широкого кола завдань і вживання заходів для забезпечення таких можливостей;
3. Аналіз думок користувачів про функціонування ЕС;
4. Розробка системи введення-виведення, що здійснює аналіз або синтез пропозицій обмеженої природної мови, що дозволяє взаємодіяти з ЕС-2 у формі, близькою до форми стандартних підручників для даної області.

Якщо ЕС успішно пройшла етап тестування, то вона може класифікуватися як промислова експертна система.

**Етап тестування.** На цьому етапі проводиться оцінка обраного способу представлення знань у ЕС в цілому. Для цього інженер знань відбирає приклади, що забезпечують перевірку всіх можливостей розробленої ЕС.

Існують такі джерела системних збоїв: тестові випадки, введення / виведення, правила виведення, стратегії управління. Наочні приклади тестів - найбільш очевидна причина відмови ЕС. У гіршому випадку тестові кейси можуть знаходитися поза предметною областю, для якої розроблено ЕС, але частіше багато тестових кейсів є занадто однорідними і не охоплюють усієї предметної області. Тому при підготовці тестових прикладів необхідно класифікувати їх за підзадачами предметної області, виділяючи стандартні випадки, визначаючи межі [9].

Найважливішою властивістю нейронних мереж є їх здатність вчитися на основі екологічних даних і як результат навчання підвищувати свою продуктивність. Продуктивність зростає з часом за певними правилами. Навчання нейронних мереж відбувається за допомогою інтерактивного процесу коригування синаптичних ваг та порогів. В ідеалі нейронна мережа отримує знання про навколишнє середовище на кожній ітерації навчального процесу.

Поняття навчання пов’язане з багатьма видами діяльності, тому важко дати цьому процесу чітке визначення. Більше того, процес навчання залежить від точки зору на нього. Це робить практично неможливим появу будь-якого точного визначення цього поняття. Наприклад, процес навчання з точки зору психолога принципово відрізняється від навчання з точки зору шкільного вчителя. З точки зору нейронної мережі, можливо, можна використовувати наступне визначення: навчання - це процес, при якому вільні параметри нейронної мережі конфігуруються шляхом моделювання середовища, в яке вбудована мережа; тип навчання визначається методом коригування цих параметрів[13]. Це визначення процесу навчання нейронної мережі припускає наступну послідовність подій:

1. У нейронну мережу надходять стимули із зовнішнього середовища.
2. В результаті першого пункту змінюються вільні параметри нейронної мережі.
3. Після зміни внутрішньої структури нейронна мережа відповідає на збудження вже іншим чином.

Вищевказаний список чітких правил вирішення проблеми навчання нейронної мережі називається алгоритмом навчання. Нескладно здогадатися, що не існує універсального алгоритму навчання, відповідного для всіх архітектур нейронних мереж. Існує лише набір засобів, представлений безліччю алгоритмів навчання, кожен з яких має свої переваги. Алгоритми навчання відрізняються один від одного способом налаштування синаптичних ваг-нейронів. Ще однією відмінною характеристикою є спосіб зв'язку навченої нейронної мережі із зовнішнім світом. У цьому контексті говорять про парадигму навчання, пов'язану з моделлю навколишнього середовища, в якому функціонує дана нейронна мережа.

Існують два концептуальних підходи до навчання нейронних мереж: навчання з учителем і навчання без учителя.

Навчання нейронної мережі з учителем припускає, що для кожного вхідного вектора з навчальної множини існує необхідне значення вихідного вектора, званого цільовим. Ці вектори утворюють навчальну пару. Ваги мережі змінюють до тих пір, поки для кожного вхідного вектора не буде отриманий прийнятний рівень відхилення вихідного вектора від цільового.

Навчання нейронної мережі без вчителя є набагато більш правдоподібною моделлю навчання з погляду біологічних коренів штучних нейронних мереж. Навчальна множина складається лише з вхідних векторів. Алгоритм навчання нейронної мережі підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення досить близьких вхідних векторів давало однакові виходи [8].

Отже, загальна схема навчання мережі матиме такий вигляд (рисунок. 2.2).

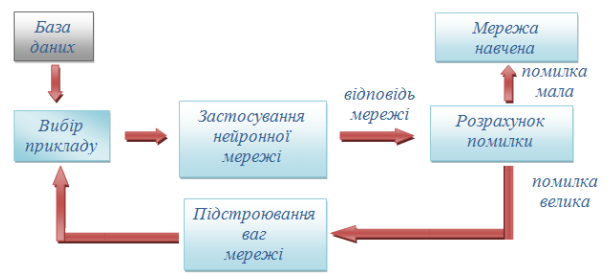
****

Рисунок. 2.2 – Загальна схема навчання мережі

Задачею класифікації називають віднесення спостереження до одного з декількох, заздалегідь відомих класів (представлених значеннями номінальної вихідної змінної). Саме її буде використано для навчання мережі.

Перед використанням нейронної мережі її необхідно навчити. Процес навчання нейронної мережі полягає в підстроюванні внутрішніх параметрів відповідно до вирішуваного завдання. Під ваговим коефіцієнтом розуміють числовий відносний показник, який характеризує ступінь складності (вагомості або важливості) певного тестового завдання. Довірчими порогами називають критерії проходження рішення через нейрон.

Алгоритм роботи нейронної мережі є ітеративним, кроки якого називають епохами або циклами. Епоха – одна ітерація в процесі навчання, на кожній ітерації в нейронну мережу подається уся навчальна вибірка (по одному спостереженню за раз) та можливість перевірки якості навчання на контрольній множині[11].

## РОЗДІЛ 3

## 3.1. Моделювання етапів розробки ЕС для заданої предметної області

На етапі формалізації було обрано інструментальним засобом програмний комплекс STATISTICA 6.0 і визначено способи подання всіх видів знань, а саме: головний біль, біль у м’язах, біль в животі, сонливість, локальні болі, відсутність апетиту та роздратованість будуть подаватися у вигляді «1» або «0», присутні або відсутні симптоми. Наступні ознаки будуть подаватися на вхід системи у такий спосіб: чутливість рецепторів – у відсотковому співвідношенні, пульс – в ударах на хвилину, температура тіла – градуси за Цельсієм.

Загальний принцип, покладений в основу формування медичних експертних систем - включення в базу знань синдромів, що відображають стан всіх основних систем органів.

## 3.2 Етап **розробки**

Щоб створити нову Таблицю даних потрібно виконати наступні дії:

1. Виберіть команду Створити в меню Файл або натисніть кнопку Створити на панелі інструментів.

2. У діалозі Створити новий Документ виберіть вкладку Таблиця.

3. Задайте Число змінних і Число спостережень.

4. У групі Положення, виберіть спосіб створення Таблиці даних: У новій Робочої книзі або В окремому вікні. Виберіть опцію У новій Робочої книзі для створення Таблиці даних в порожній Робочої книзі. Виберіть опцію В окремому вікні, щоб створити порожню таблицю даних в новому вікні.

5. Натисніть кнопку OK.

## 3.2.1 **Етапи створення нової мережі.** Спочатку оберіть в меню *Аналіз → Нейронні мережі* (Рисунок.3.1).

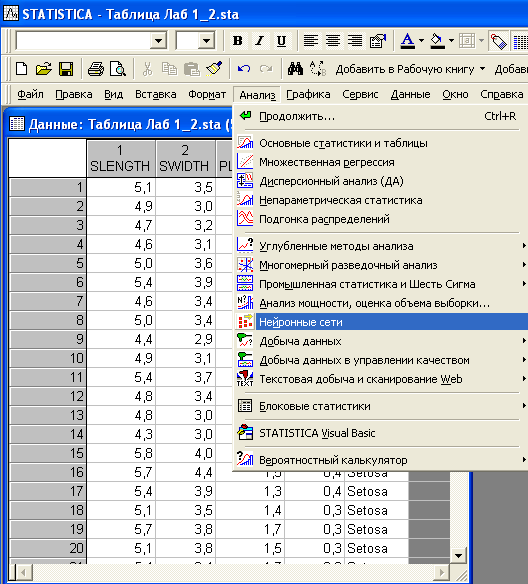


Рисунок 3.1 – Створення нової нейромережі

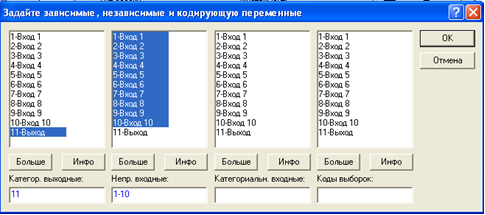


Рисунок 3.2 – Встановлення типу змінних

## 3.2.2 Робота з таблицею даних. Заголовок вікна. В області Тема вікна відображається назва Таблиці даних разом зі стандартним розширенням, прийнятим для Таблиць даних (.sta). Якщо Таблиця даних є вихідною Таблицею даних, то в області Заголовок вікна також міститься число змінних і спостережень (наприклад, 25п \* 50н). На малюнку області Тема вікна відображається текст Data: Adstudy.sta (25v by 50c). \

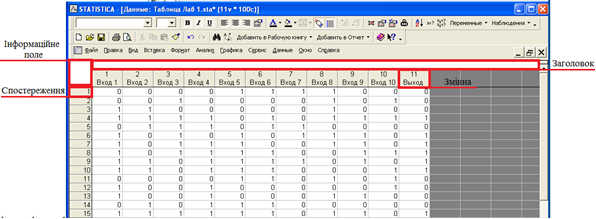


Рисунок 3.4 – Таблиця в програмному комплексі STATISTICA

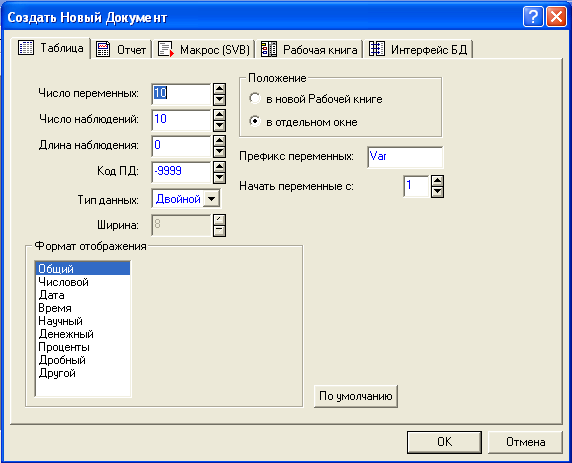


Рисунок 3.5 – Створенная таблиці

Двічі натисніть в області Тема, зверху вікна над іменами змінних, для того, щоб ввести або змінити текстову інформацію. Для виділення самої області Тема (наприклад, у разі форматування) натисніть у верхньому лівому кутку (курсор миші при цьому набуває вигляду).

**Імена спостережень.** Ці осередки розташовані ліворуч від вікна таблиці даних, і містять імена для кожного спостереження. Щоб ввести або редагувати текст, двічі клацніть на будь-яке ім'я спостереження. Щоб виділити самі імена спостережень (наприклад, у випадку форматування), клацніть один раз на лівій стороні імені спостереження (з’явиться вказівник миші). Щоб вибрати всю лінію спостереження (для редагування), клацніть один раз у середній або правій частині Імені спостереження (з’явиться вказівник миші). Щоб вибрати блок Імена спостережень (без виділення самих рядків), виберіть Вибрати лише імена спостереження в контекстному меню Таблиці даних. Щоб автоматично регулювати ширину полів Ім'я спостереження, двічі клацніть правою межею будь-якого поля Ім'я спостереження (з’явиться вказівник миші). На малюнку вище клітини Назви спостереження містять перші ініціали та прізвища респондентів. Зверніть увагу, що імена спостережень не обов’язково присутні у вікні Таблиці даних, і ви можете їх приховати (вимкніть параметр Показувати імена спостережень у меню Перегляд). У такому випадку, якщо вони не існують, відображаються лише номери спостережень [9].

**Імена змінних.** Ці комірки, розташовані вгорі кожного стовпця, містять імена змінних. Щоб переглянути специфікації окремої змінної, двічі клацніть у полі Ім'я змінної. Щоб вибрати саме поле Ім'я змінної (наприклад, у випадку форматування), клацніть один раз у верхній частині поля Ім'я змінної (замість цього з’явиться вказівник миші). Щоб вибрати цілий стовпець змінної (наприклад, під час редагування), клацніть один раз внизу поля Ім'я змінної (з’явиться вказівник миші). Щоб автоматично регулювати ширину стовпця, двічі клацніть правою межею поля Ім'я змінної (з’явиться вказівник миші). На малюнку вище перші дві клітинки імені змінної містять текст Gender та Advert [9].

## 3.3 Навчання нейронної мережі

Після створення відповідної мережі необхідно провести процедуру навчання. На кожній ітерації (епохи) в нейронну мережу подається вся навчальна вибірка (по одному спостереженню за один раз).

Вихідні значення порівнюються з відповідними вихідними значеннями в навчальній вибірці, а помилка між отриманим і потрібним результатами використовується для коригування ваг в нейронної мережі для зменшення помилки [10].

Для навчання мережі необхідно виконати наступні кроки. Навчання з використанням зворотного поширення. Діалог Багатошаровий персептрон - навчання реалізує навчання СНТІ. Докладний загальний алгоритм буде розглянуто пізніше, а зараз ми зупинимося на простому алгоритмі.

Перейдіть на вкладку Інтерактивний і виберіть опцію Інтерактивне навчання. Це дозволить відображати графік під час навчання.

Натисніть кнопку OK, щоб запустити алгоритм навчання. Після цього буде відображено діалог Йде навчання, а на графіку будуть відображатися дві криві. Перша крива позначає помилку на навчальній вибірці (дані, які використовуються для оптимізації ваг нейронної мережі), а друга - помилку на контрольній вибірці (це дозволяє виконувати крос-перевірку під час навчання).

Після зупинки навчання натисніть кнопку Додатково, а потім кнопку OK в діалозі Розширене навчання лінійних елементів для побудови додаткових епох навчання. Ви можете повторити цей процес до тих пір, поки не будуть досягнуті бажані результати. Потім натисніть першу кнопку OK в діалозі Йде навчання.

Після цього з'явиться діалог Результати, також як після запуску Майстра рішень. Використовуйте кнопки Описові статистики і Прогнози для перевірки продуктивності мережі, а також порівняйте продуктивність отриманої мережі з продуктивністю попередніх мереж, використовуючи підсумкові статистики у верхній частині діалогу Результати. Після перевірки мережі натисніть кнопку OK, щоб зберегти її в наборі мереж.

Перенавчання існуючої мережі. Замість послідовного створення нових мереж, які швидко заповнюють файл, ми перенавчитися створену нейронну мережу (це можливо тільки в тому випадку, якщо ви використовуєте таку ж нейромережевому архітектуру - тип, вхідні і вихідні змінні, число прихованих шарів і елементів, і т.д. ).

Виберіть нейронну мережу, яку хочете перенавчити. Це можна зробити на вкладці Мережі / Ансамблі на Стартовою панелі, а потім виберіть перенавчитися мережу в списку Вибрати аналіз на вкладці Додатково.

Оптимізація продуктивності навчання. Продуктивність алгоритму зворотного поширення залежить від декількох параметрів, які доступні в діалозі Багатошаровий персептрон - навчання. Стандартні значення можна використовувати в більшості відомих задач, однак, ви завжди можете змінити їх, адаптувавши для завдання про іриси.

**Епохи.** У цьому полі визначається число розглянутих епох при кожному натисканні кнопки OK. За замовчуванням прийнято невелике значення 100; збільште це значення до 500 [10].

**Швидкість навчання.** Висока швидкість навчання дозволяє швидко виконати весь алгоритм, проте в деяких завданнях можуть бути отримані нестабільні результати. У нашій задачі можна використовувати значення 0.01 [10].

З такими заданими параметрами STATISTICA Нейронні мережі нормально вирішить типову задачу менш ніж за тисячу ітерацій. Навчання нейронної мережі з крос-ентропійною функцією помилки (і відповідними функціями активації вихідного шару) еквівалентно оптимізації мережі за методом максимальної правдоподібності в припущенні, що розподіл даних належить експоненціальному сімейству розподілу (в т.ч. і нормальний розподіл). Довірчі рівні, створені подібної мережею, можна інтерпретувати як ймовірності[13].

Щоб навчити нейронну мережу класифікації, використовуючи квадратичний метод, необхідно перервати всі запущені аналізи, запустити Конструктор мереж, вибрати нейронну мережу Багатошаровий персептрон, а потім перейти на вкладку Елементи. Виберіть опцію Сума квадратів в групі Функція класифікації та натисніть кнопку OK.

Використовуючи квадратчну функцію, можна встановити більш високу швидкість навчання. Встановіть в поле Швидкість навчання значення 0.5 і зменшіть число Епох до 100.

**Момент.** Цей параметр зазвичай покращує продуктивність за рахунок прискорення навчання, коли існують невеликі зміни помилки. Значення цього параметра завжди повинні знаходитися в інтервалі [0.0,1.0) (тобто більше або дорівнює 0.0 і строго менше 1.0). Рекомендується доповнювати високу швидкість навчання низьким моментом, і навпаки. В даному випадку, припустимо великий момент - введемо значення 0.6 [10].

**Зауваження:** *Програмний комплекс STATISTICA може змінювати Швидкість навчання та/або Момент для кожної епохи. При цьому значення можуть варіюватися від початкових значень, заданих в лівих полях на вкладці BP, до кінцевих значень, заданих в відповідних правих полях. Наприклад, ви можете зменшувати швидкість навчання під час самого процесу навчання. Для цього необхідно вибрати опцію Коригувати швидкість навчання і момент. Якщо ця опція не вибрана, то STATISTICA буде використовувати фіксовані параметри, задані в лівих полях.*

Спуск по зв'язаних градієнтах є альтернативним навчальним алгоритмом, який зазвичай працює швидше, ніж алгоритм зворотного поширення. Другою перевагою є те, що не потрібно вибирати швидкість навчання і момент, тобто використовувати цей алгоритм набагато простіше.

Натисніть кнопку OK, щоб запустити алгоритм. Після цього можна очікувати наступного:

Коли навчання перемикається на спуск по зв'язаних градієнтам, то в процесі навчання стає значно менше перешкод. Це відбувається через те, що в алгоритмі зворотного поширення ми використовуємо опцію перемішування, яка додає значне число перешкод. Однак, завдяки цьому, алгоритм зворотного поширення швидше досягає локального мінімуму.

Різні епохи спуску по зв'язаних градієнтам займають більше часу, ніж епохи зворотного поширення, тому цей алгоритм спочатку може здатися більш повільним. Однак, рівень помилки зменшується значно швидше, тому спуск по зв'язаних градієнтам займає менше часу під час навчання.

Після зупинки навчання натисніть кнопку Додатково, а потім кнопку OK в діалозі Розширене навчання лінійних елементів для побудови[9].

## 3.4 Режим інтерактивного навчання

Програмний комплекс STATISTICA може відображати рівень поточної помилки на графіку, дозволяючи таким чином інтерактивно спостерігати за процесом навчання.

Виберіть вкладку Інтерактивний вікна - Багатошаровий персептрон - навчання для доступу до опцій, які належать до інтерактивного навчання. При інтерактивному навчанні вікно виводиться під час навчання, показуючи динаміку функції помилки (дивіться вікно Йде навчання), яке включає опції для ранньої зупинки навчання і для розширеного навчання з додатковими епохами тих же самих або інших алгоритмів навчання.

**Інтерактивне навчання (відображати графік).** Виберіть цю опцію, щоб включити інтерактивне навчання. При натисканні кнопки ОК для виконання алгоритму навчання виводиться вікно Йде навчання.

**Зберігати попередній графік (для порівняння).** Якщо опція вибрана, графік не очищається на початку навчання, дозволяючи порівняти результати Вашого прогону навчання з попередніми прогонами.

**Автоматично закривати графік після закінчення навчання.** За замовчуванням, коли закінчується інтерактивне навчання, вікно Йде навчання залишається видимим, так що Ви можете розширити навчання або переслати графік в робочу книгу результатів. Якщо опція вибрана, вікно автоматично закривається після закінчення навчання. Виберіть цю опцію, якщо хочете розглянути графік для цілей зворотного зв'язку, але не хочете продовжувати поточне навчання.

**Графіки.** Вкажіть, які графіки показувати. За замовчуванням виводяться графіки та контрольної помилки, і помилки навчання; однак, Ви можете вибрати висновок тільки однією з них, щоб зменшити захаращеність графіка.

**Період виведення (в епохах).** Визначити період (в епохах) виведення помилки. Якщо Ви очікуєте великого числа епох, задайте інтервал виведення більше одиниці, щоб скоротити вимоги до пам'яті та час промальовування графіка.

У діалозі йде навчання знаходяться три вкладки: Графік, Динамічні опції і Статичні опції.

У цьому діалозі відображається графік навчання та / або значення помилки для заданої вибірки. Цей графік оновлюється в режимі реального часу (див. Опис вкладки Графік). Ви можете перервати навчання в будь-який момент (закінчивши навчання і зберігши нейронну мережу, або скасувавши навчання), а також збільшити період навчання на кілька епох. Крім цього, ви можете визначити заголовок графік, вставити легенду і мітки ліній (див. Опис вкладки Динамічні опції) на графік результатів.

У нижньому правому куті діалогу відображається рядок станів і час роботи. Також відображається відсоток зайнятості центрального процесора.

**Готово.** Під час роботи алгоритму відображається кнопка Готово. Натисніть цю кнопку, щоб зупинити навчання. Після цього буде невелика затримка, під час якої програма завершує навчання для поточної доби. Після завершення алгоритму кнопка Готово замінюється на кнопку OK. Натисніть кнопку OK, щоб успішно завершити алгоритм. Після цього буде відображено діалог Результати.

**OK.** Кнопка OK з'являється після закінчення роботи алгоритму. Натисніть цю кнопку, щоб успішно завершити алгоритм і скопіювати отриманий графік в Графік STATISTICA.

**Додатково.** Натисніть цю кнопку, щоб задати додаткові епохи і / або змінити параметри алгоритму навчання. При цьому відображається спеціальний розширений діалог. При натисканні кнопки OK в цьому діалозі знову відображається діалог Йде навчання.

**Скасування.** Натисніть кнопку Скасувати, щоб перервати навчання і повернутися в діалог Навчання, з якого була запущена дана процедура. При натисканні кнопки Скасування буде відображено підтвердження про скасування навчання. Щоб підтвердити це, натисніть кнопку OK в діалогу Скасування навчання.

## 3.5 Методика рішення задачі

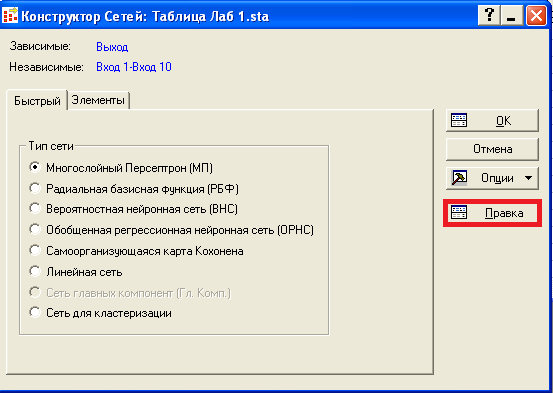


Рисунок 3.6 – Діалогове вікно *Конструктора мереж*

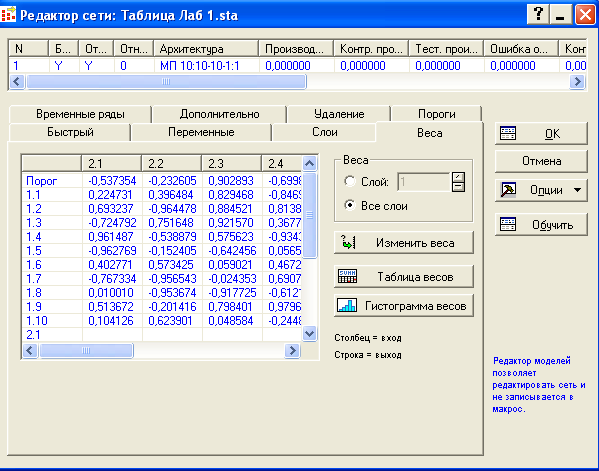


Рисунок 3.7 – Діалогове вікно *Редактора мереж*

Дано завдання створити вибірку, що складається з 10 компонентів, кожен з яких може бути нулем або одиницею:

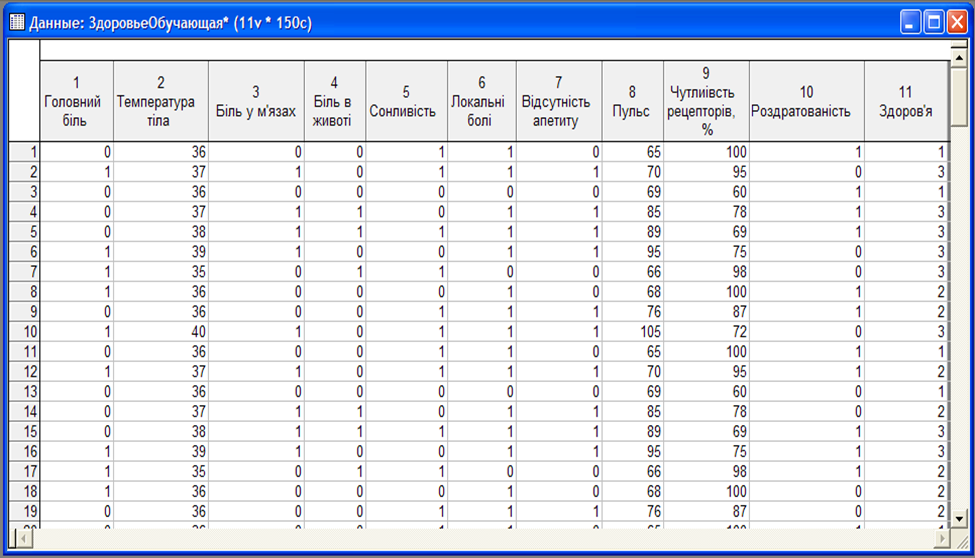
Рисунок. 3.8 – Створення таблиці з 10 компонентів

Рисунок 3.9 – Таблиця з 10 компонентів

## 3.5.1 Вибір та аналіз нейроархітектури:

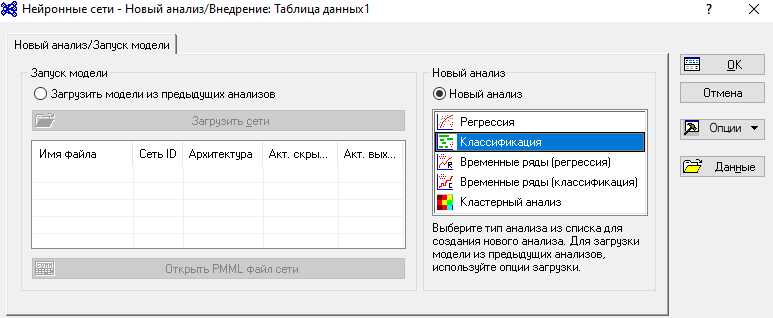


Рисунок 3.10 – Вибір нейронної мережі для задачі «Класифікації»

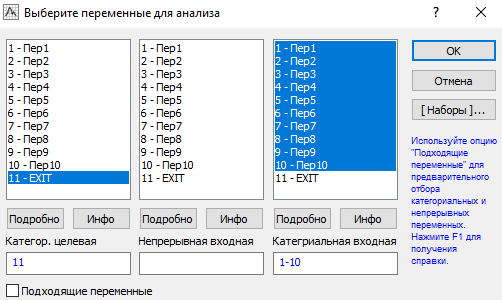


Рисунок 3.11 – Вибір вхідних та вихідних даних для навчання нейронної мережі за методом класифікації

## 3.5.2 Аналіз результатів навченої мережі:

Рисунок 3.12 – Вибір типу мережі

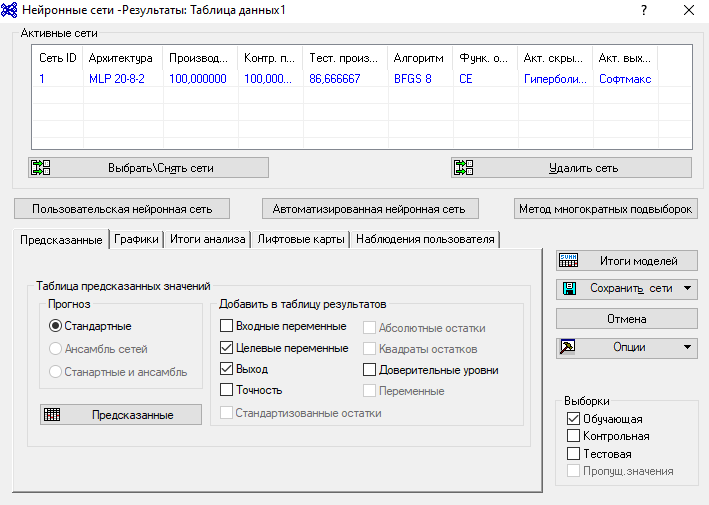


Рисунок 3.13 – Навчена нейронна мережа

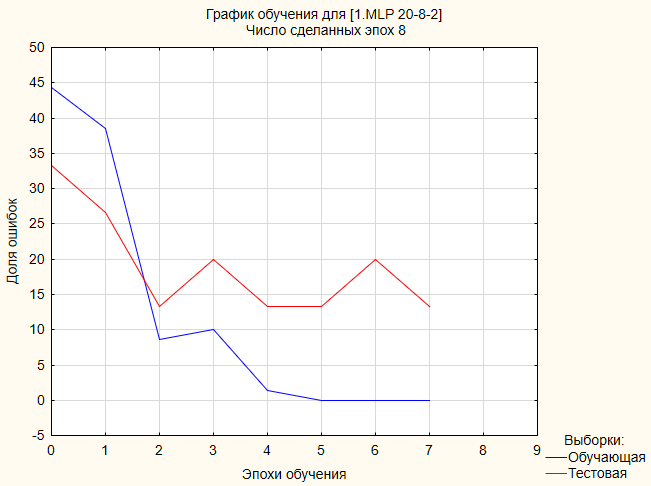


Рисунок 3.14 – Графік помилок

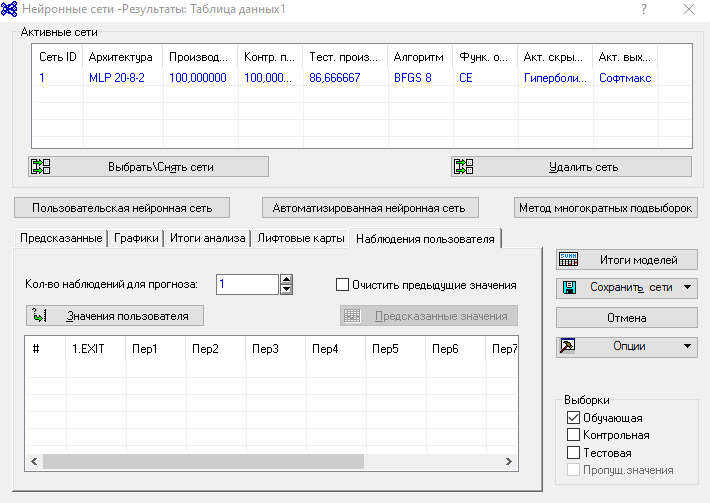


Рисунок 3.15 – Вкладка користувацьких передбачень

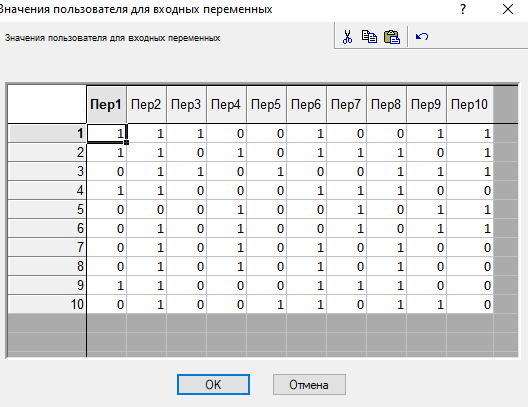


Рисунок 3.16 – Введення даних для власного передбачення

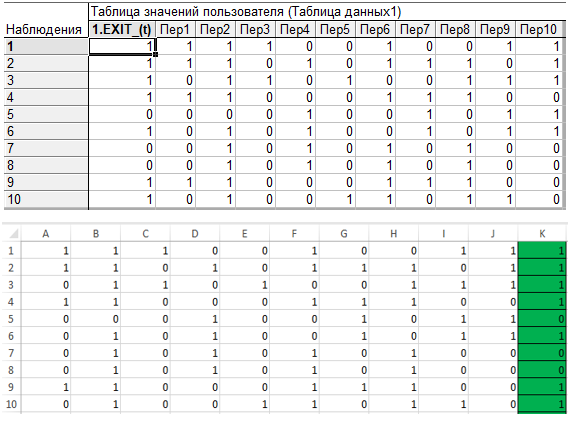


Рисунок 3.17 – Порівняння передбачених та реальних результатів

В результаті аналізу графіку помилок та користувацьких передбачень помилок не виявлено. Тому, ми можемо сказати, що наша мережа повністю обучилась.

## РОЗДІЛ 4

## 4.1 РОЗВЯЗАННЯ ЗАДАЧІ

Отже ми маємо вибірку Здоров’я людини. Вона містить в собі 10 стовпців вхідних даних:

1. Головний біль
2. Температура тіла
3. Біль у м’язах
4. Біль в животі
5. Сонливість
6. Локальні болі
7. Відсутність апетиту
8. Пульс
9. Чутливість рецепторів, %
10. Роздратованість

Стовбець вихідних даних - Здоров'я. Він визначається заданими вище факторами та містить в собі 3 класи, за якими класифікуються дані, а саме:

1. Гарний стан здоров’я

2. Задовільний стан

3. Незадовільний стан

Задано 200 спостережень. Необхідно розробити модель класифікатора, досягнувши критерію мінімальних помилок та максимальної продуктивності, а також досягти правильної класифікації на тестових даних.

Для розв’язання поставленої задачі запускаємо вибірку у програмному середовищі STATISTICA 6.0. На Рисунку 4.1 зображено фрагмент навчальної вибірки.

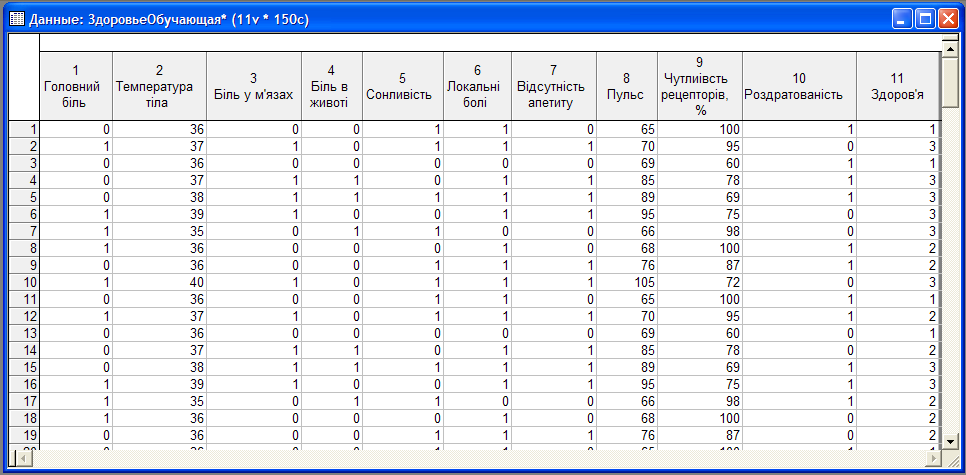


Рисунок 4.1 – Фрагмент навчальної вибірки

Запускаємо модуль Нейронні мережі STATISTICA. На стартовому вікні вибираємо тип завдання: *Классификация.* За інструмент обираємо *Мастер решений*

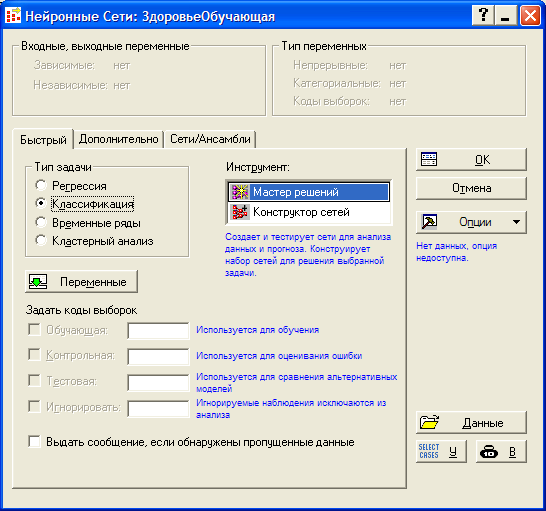


Рисунок 4.2 – Вікно *Нейронные сети*

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 10 вхідних змінних. У лівому вікні виберемо безперервну вихідну змінну, - це змінна «Здоров’я» з номером 11. Виберемо вхідні змінні - це незалежні змінні, які передбачають відгук.

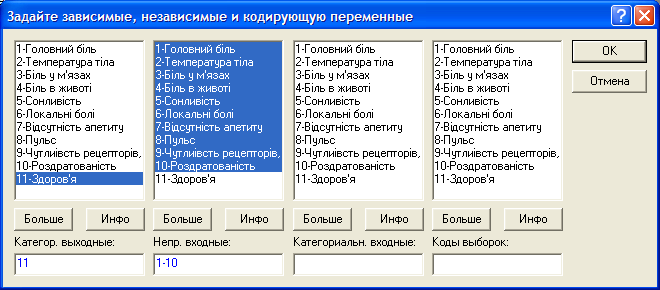


Рисунок 4.3 – Вікно задання змінних

Далі на вкладці *Быстрый* обираємо *Длительность анализа*: *N сетей* – 10, *Сохранять сетей* – 5.

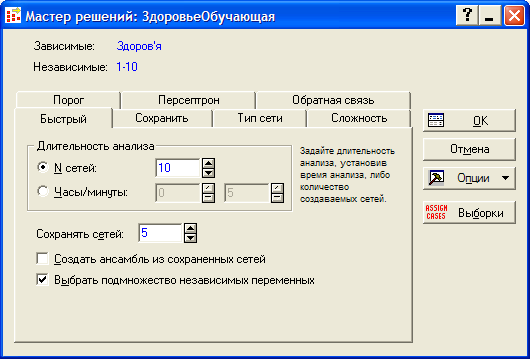


Рисунок 4.4 – Вікно *Мастер решений*

Далі натискаємо на *Выборки*.

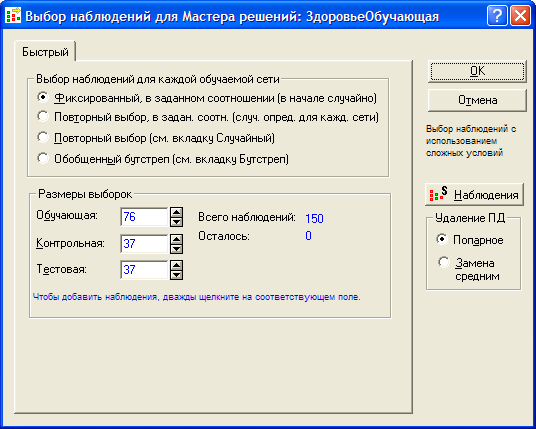


Рисунок 4.5 – Вікно *Формирование выборок*

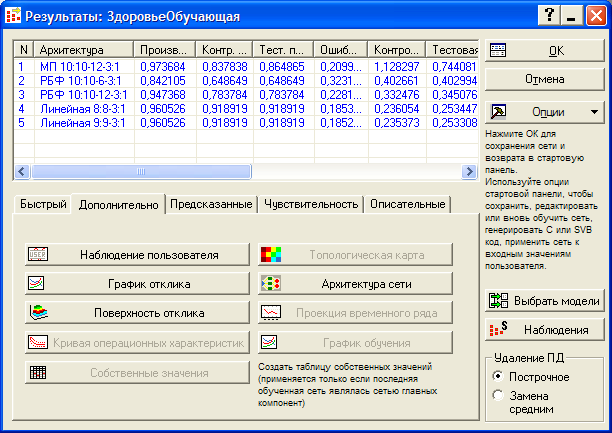
Розміри вибірок обираємо: Навчальна – 76, Контрольна – 37, Тестова – 37.

Рисунок 4.6 – Результати навчання мережі

Отже, отримали 5 найкращих мереж. На кладці *Быстрый* обираємо *Итоги сетей*. Бачимо докладний результат.

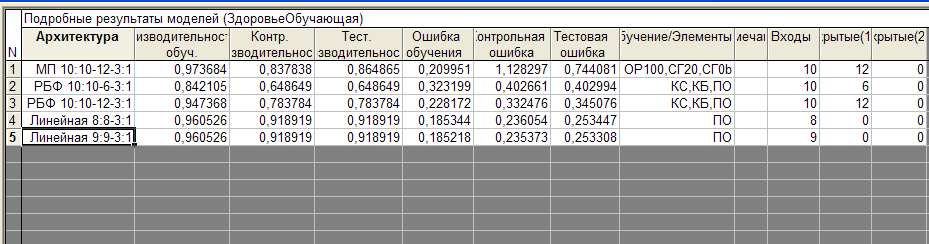


Рисунок 4.7 – Докладні результати моделей

Отже, мережа *Линейная 9:9-3:1* дала найкращий результат, а саме:

* Продуктивність – 0,960526
* Помилки в діапазоні 0,19-0,26.

Зберігаємо модель.

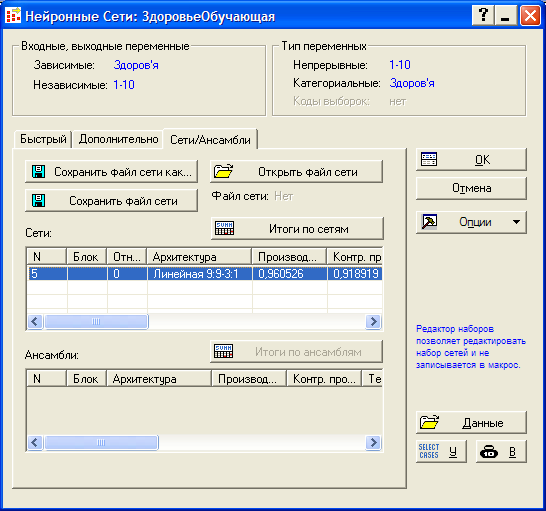


Рисунок 4.8 – Збереження моделі

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 збережену модель.

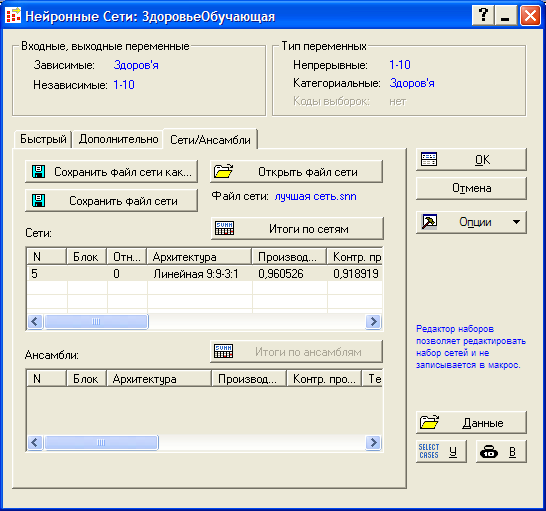


Рисунок 4.9 – Вікно *Нейронные сети*

У вкладці *Дополнительно* обираємо *Запустить модель.*

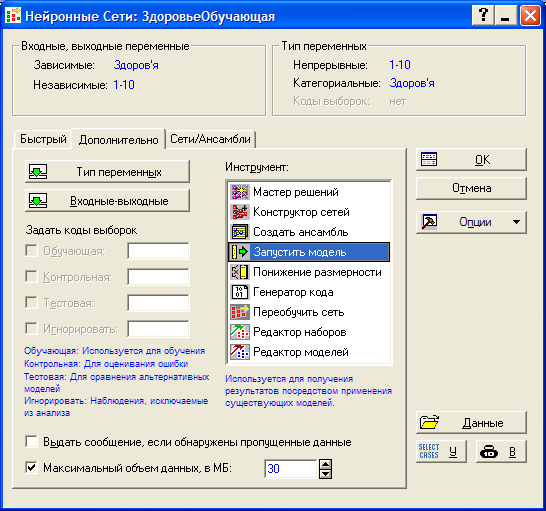


Рисунок 4.10 – Вікно *Нейронные сети*

Наступним кроком відкриваємо вкладку *Дополнительно* та натискаємо *Наблюдения пользователя*.

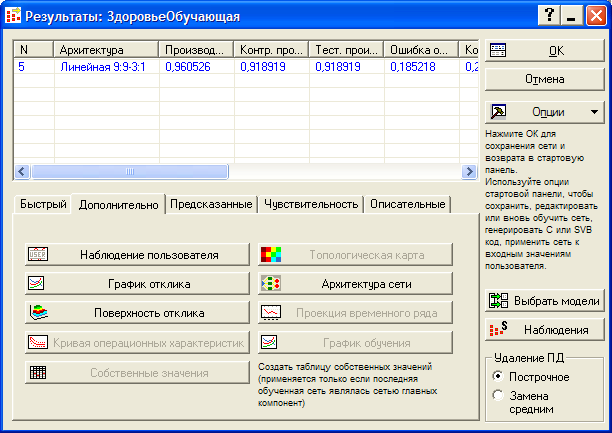


Рисунок 4.11 – Результати мережі

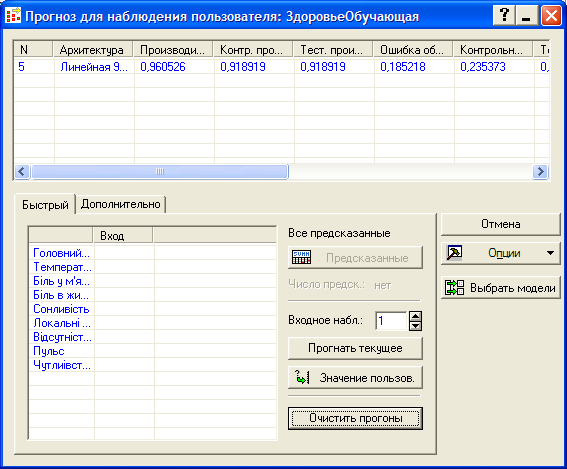


Рисунок 4.12 – Вікно *Наблюдения пользователя*

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

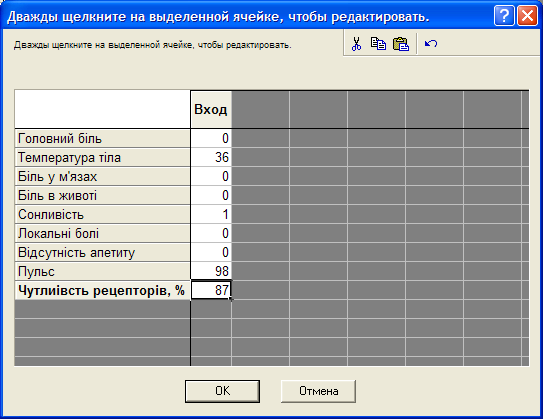


Рисунок 4.13 – Вікно *Значения пользователя*

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

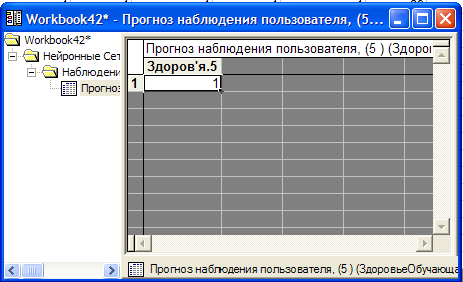


Рисунок 4.14 – Вікно *Прогноз наблюдения пользователя*

Отже, система автоматично класифікувала дані до 1 класу, що відповідає дійсності. Для перевірки проведемо ще декілька перевірок.

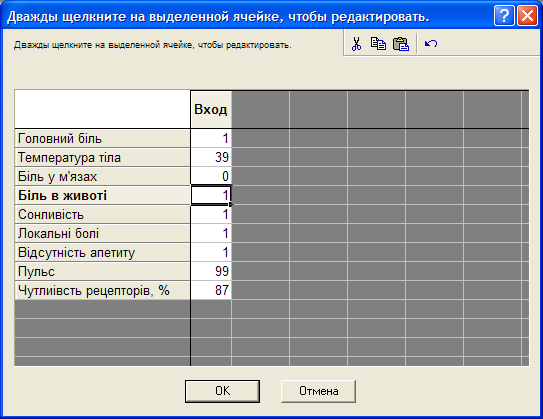


Рисунок 4.15 – Вікно *Значения пользователя*

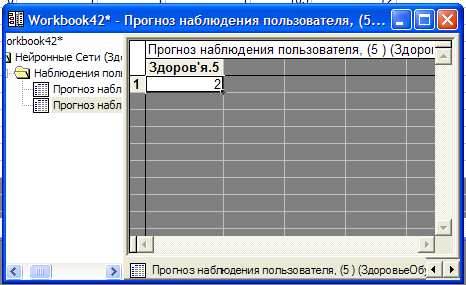


Рисунок 4.16 – Вікно *Прогноз наблюдения пользователя*

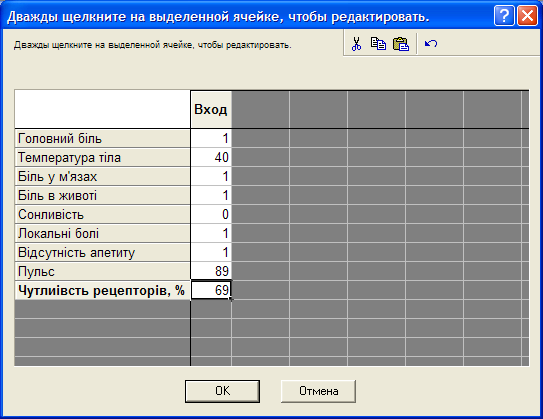


Рисунок 4.17 – Вікно *Значения пользователя*

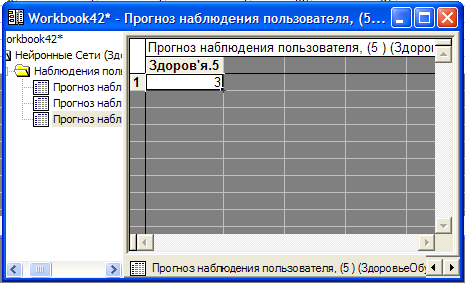


Рисунок 4.18 – Вікно *Прогноз наблюдения пользователя*

Отже, розглянувши декілька прикладів на нових даних, можна сказати, що система класифікує дані вірно, а отже, модель дійсно гарно навчилася та може бути використана для автоматичного визначення стану здоров’я за даними признаками.

## ВИСНОВКИ

В ході виконання роботи було розглянуто етапи проектування експертних систем з практичним обґрунтуванням відповідно до заданої предметної області – визначення стану здоров’я за ознаками, котрі людина може самостійно визначати.

Був синтезований ансамбль нейромережевих моделей реалізуючих класифікацію станів об'єкта дослідження;

створений програмний код кінцевої моделі як самостійний додаток для скринінг-аналізу стану здоров'я;

Необхідна адекватність моделей досягнута тестуванням на вибірці прецедентів які не брали участь в процесі навчання мереж.

В роботі було проведено аналіз для навчальної вибірки та за допомогою майстра рішень знайдено кращу модель:

– Продуктивність – 0,960526

– Помилки в діапазоні 0,19-0,26.

Після збереження навченої моделі було проведено перевірку на правильність класифікації станів за ознаками. Інструментарії програмного засобу STATISTICA дозволяють це робити досить зручно.

Також, було досягнено поставленої мети, а саме: дослідження нейромережевих технологій визначення стану самопочуття організму.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Експертні системи. Навчальні матеріали онлайн [Електронний ресурс]–Режим доступу: http://pidruchniki.com/10811007/informatika/ekspertni\_sistemi
2. Рутковский Лешек. Методы и технологии искусственного интеллекта / Пер. с польск. И. Д. Рудинского. — М.: Горячая линия — Телеком, 2010. — 520 с.
3. Сетлак Г. Интеллектуальные системы поддержки принятия решений. — К.: Логос, 2004. — 179 с
4. Д. Джарратано, Г. Райлі. Експертні системи. Принципи розробки та програмування. Вид. Вільямс, 2006.
5. Альошин  С.П. Свідоцтво про реєстрацію авторського права на твір № 50846. Автоматизована нейромережева модель формули збалансованого  харчування //  Альошин  С.П.,   Ляхов  О.Л., Бородіна О.О.   Державна  служба інтелектуальної власності України. Дата реєстрації 21.08. 2013.
6. Моїсеєв В.Б. Подання знань в інтелектуальних системах. Інформатика і освіта,.№ 2, 2003 р. с. 84-91.
7. Гасанова, И. А. Искусственный интеллект в клинической медицине и стоматологии / И. А. Гасанова, В. А. Парийский, А. А. Грибков // Тенденции развития науки и образования. – 2018. – Т. 38, № 4. – С. 17-21
8. Исхакова, А. Ф. Применение искусственного интеллекта / А. Ф. Исхакова // Вестник современных исследований. - 2018. - № 9.3 (24) .- С. 261-262.
9. Снитюк В.Є. Прогнозування. Моделі, методи, алгоритми. – К.: Маклаут, 2008. – 364 с.
10. Глибовець М.М., Олецький О.В. Штучний інтелект: Підручн. для студ. вищ. навч. закладів, що навчаються за спец. «Компютерні науки» та «Прикладна математика». – К.: Вид. дім «КМ Академія», 2002. – 366 с.
11. Штучні нейромережі. Навчальні матеріали онлайн [Електронний ресурс] – Режим доступу:
12. Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирования, 4-е издание.: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2007. – 1152 с.
13. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с
14. Вінер Н. Кібернетика. — М.: Наука. — 1983. — 356 с.
15. Салютін Ш.М. Штучний інтелект. — М.: Думка, 1995. — 200 с.

## ДОДАТОК А

**Лістинг програми**

Option Base 1

Sub Main

Dim newanalysis As Analysis

Set newanalysis = Analysis (scNeuralNetworks, ActiveDataSet)

With newanalysis.Dialog

.VariablesAndTypes = "1-10 | 11"

.Variables = "11 | 1-10"

.TypeOfAnalysis = scSNNIntelligentProblemSolver

.IssueMissingDataWarning = False

.ApplyMemoryLimit = True

.MemoryLimit = 30

End With

newanalysis.Run

With newanalysis.Dialog

.IPSRunTimeBasedOnNetworksTested = True

.NoOfNetworksTestedByIPS = 10

.NoOfNetworksRetainedByIPS = 5

.FormEnsemble = False

.SelectSubsetOfIndependents = True

.BalancePerformance = True

.IncreaseFileSize = True

.SaveCopyToFile = False

.LinearNetwork = True

.PNNNetwork = False

.RBFNetwork = True

.MLP3Network = True

.MLP4Network = False

.RBFMinComplexity = 1

.RBFMaxComplexity = 38

.MLP3MinComplexity = 1

.MLP3MaxComplexity = 12

.AssignToHighestConfidence = True

.TreatAsTimeSeries = False

.ClassificationEntropyBased = True

.ClassificationSumSquaredLogistic = False

.FeedbackImprovedNetworksOnly = True

.GenerateSpreadsheet = False

.DisplayTerminalMessage = False

.CopyTerminalMessageToClipboard = True

End With

Set optionsdialog = newanalysis.Dialog.GetUserCaseSelection

With optionsdialog

.SelectSpecifiedCases = True

.UserSpecifiedCases = "1-150"

End With

newanalysis.Dialog.SetUserCaseSelection optionsdialog

Set optionsdialog = Nothing

Set optionsdialog = newanalysis.Dialog.GetSampling

With optionsdialog

.CasewiseDeletionOfMD = True

.FixSelectionOfSubsets = True

.TrainingSubsetSizeSimple = 76

.SelectionSubsetSizeSimple = 37

.TestSubsetSizeSimple = 37

.RandomFixAllSubsets = True

.AssignmentOfSubsetsRandomOnce = True

.TrainingSubsetSizeRandom = 76

.SelectionSubsetSizeRandom = 37

.TestSubsetSizeRandom = 37

.TrainingSubsetSizeBootstrap = 100

.TestSubsetSizeBootstrap = 50

End With

newanalysis.Dialog.SetSampling optionsdialog

Set optionsdialog = Nothing

newanalysis.Run

With newanalysis.Dialog

.UseOverallSubsets = True

.GenerateSummaryStatistics = True

.GenerateConfusionMatrix = True

.ShowPredictions = True

.ShowConfidenceLevels = False

.ShowCodebookVector = False

.ShowWiningNeuron = False

.IncludeObservedInPredictionsSpreadsheet = True

.IncludeIndependentsInPredictionsSpreadsheet = False

.IncludeSubsetVariableInPredictionsSpreadsheet = False

.IncludeUserVariablesInPredictionsSpreadsheet = False

.ShowRawResiduals = True

.ShowResidualsSquared = False

.ShowAbsoluteResiduals = False

.ShowStandardResiduals = False

.IncludePredictionsInResidualsSpreadsheet = True

.IncludeObservedInResidualsSpreadsheet = False

.IncludeIndependentsInResidualsSpreadsheet = False

.IncludeSubsetVariableInResidualsSpreadsheet = False

.IncludeUserVariablesInResidualsSpreadsheet = False

.ShowRatioAndRankingSensitivityMetrics = True

.SelectXaxisVariable = 1

.SelectYaxisVariable = 0

.CasewiseDeletionOfMD = True

End With

Set optionsdialog = newanalysis.Dialog.GetUserCaseSelection

With optionsdialog

.SelectSpecifiedCases = True

.UserSpecifiedCases = "1-150"

End With

newanalysis.Dialog.SetUserCaseSelection optionsdialog

Set optionsdialog = Nothing

Set optionsdialog = newanalysis.Dialog.GetMultipleModelSelection

With optionsdialog

.ShowOnlyStandaloneNetworks = True

.SelectTheEnsembleItself = True

.SelectedModels = "1-5 "

End With

newanalysis.Dialog.SetMultipleModelSelection optionsdialog

Set optionsdialog = Nothing

newanalysis.RouteOutput(newanalysis.Dialog.PrintModelSummarySpreadsheet).Visible = True

Set optionsdialog = newanalysis.Dialog.GetMultipleModelSelection

With optionsdialog

.SelectedModels = "5 "

End With

newanalysis.Dialog.SetMultipleModelSelection optionsdialog

Set optionsdialog = Nothing

newanalysis.Run

With newanalysis.Dialog

.VariablesAndTypes = "1-10 | 11"

.Variables = "11 | 1-10"

.TypeOfAnalysis = 7

.NetworkListView = " 5 "

.IssueMissingDataWarning = False

.ApplyMemoryLimit = True

.MemoryLimit = 30

End With

newanalysis.Run

newanalysis.Run

With newanalysis.Dialog

.VariablesAndTypes = "1-10 | 11"

.Variables = "11 | 1-10"

.TypeOfAnalysis = 7

.NetworkListView = " 5 "

.IssueMissingDataWarning = False

.ApplyMemoryLimit = True

.MemoryLimit = 30

End With

newanalysis.Dialog.SaveNetworkFile "C:\Documents and Settings\User\Рабочий стол\Здоровье\лучшая сеть.snn"

newanalysis.Dialog.OpenNetworkFile "C:\Documents and Settings\User\Рабочий стол\Здоровье\лучшая сеть.snn"

With newanalysis.Dialog

.TypeOfAnalysis = scSNNRunExistingModel

End With

newanalysis.Run

With newanalysis.Dialog

.UseOverallSubsets = True

.GenerateSummaryStatistics = True

.GenerateConfusionMatrix = True

.ShowPredictions = True

.ShowConfidenceLevels = False

.IncludeObservedInPredictionsSpreadsheet = True

.IncludeIndependentsInPredictionsSpreadsheet = False

.IncludeSubsetVariableInPredictionsSpreadsheet = False

.IncludeUserVariablesInPredictionsSpreadsheet = False

.ShowRawResiduals = True

.ShowResidualsSquared = False

.ShowAbsoluteResiduals = False

.ShowStandardResiduals = False

.IncludePredictionsInResidualsSpreadsheet = True

.IncludeObservedInResidualsSpreadsheet = False

.IncludeIndependentsInResidualsSpreadsheet = False

.IncludeSubsetVariableInResidualsSpreadsheet = False

.IncludeUserVariablesInResidualsSpreadsheet = False

.ShowRatioAndRankingSensitivityMetrics = True

.SelectXaxisVariable = 1

.SelectYaxisVariable = 0

.CasewiseDeletionOfMD = True

End With

Set optionsdialog = newanalysis.Dialog.GetUserCaseSelection

With optionsdialog

.SelectSpecifiedCases = True

.UserSpecifiedCases = "1-150"

End With

newanalysis.Dialog.SetUserCaseSelection optionsdialog

Set optionsdialog = Nothing

Set optionsdialog = newanalysis.Dialog.GetMultipleModelSelection

With optionsdialog

.ShowOnlyStandaloneNetworks = True

.SelectTheEnsembleItself = True

.SelectedModels = "5 "

End With

newanalysis.Dialog.SetMultipleModelSelection optionsdialog

Set optionsdialog = Nothing

newanalysis.Dialog.GotoUserDefinedCase

With newanalysis.Dialog

.UseInputValue = 1

.IncludeStandardPredictionsInPredictionsSpreadsheet = True

.IncludeConfidenceLevelsInPredictionsSpreadsheet = False

.IncludeInputsInPredictionsSpreadsheet = False

.RetainMultiplePredictionsInSpreadsheet = True

.RunUserDefinedInput = "(0,000000)" + vbLf + \_

"(36,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(98,000000)" + vbLf + \_

"(87,000000)"

End With

Set optionsdialog = newanalysis.Dialog.GetSelectModelsForExecution

With optionsdialog

.ShowOnlyStandaloneNetworks = True

.SelectTheEnsembleItself = True

.SelectedModels = "5 "

End With

newanalysis.Dialog.SetSelectModelsForExecution optionsdialog

Set optionsdialog = Nothing

newanalysis.RouteOutput(newanalysis.Dialog.PrintPredictionsSpreadsheet).Visible = True

With newanalysis.Dialog

.RunUserDefinedInput = "(1,000000)" + vbLf + \_

"(37,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(60,000000)" + vbLf + \_

"(98,000000)"

End With

newanalysis.RouteOutput(newanalysis.Dialog.PrintPredictionsSpreadsheet).Visible = True

With newanalysis.Dialog

.RunUserDefinedInput = "(1,000000)" + vbLf + \_

"(38,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(98,000000)" + vbLf + \_

"(78,000000)"

End With

newanalysis.RouteOutput(newanalysis.Dialog.PrintPredictionsSpreadsheet).Visible = True

With newanalysis.Dialog

.RunUserDefinedInput = "(0,000000)" + vbLf + \_

"(37,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(87,000000)" + vbLf + \_

"(99,000000)"

End With

newanalysis.RouteOutput(newanalysis.Dialog.PrintPredictionsSpreadsheet).Visible = True

With newanalysis.Dialog

.RunUserDefinedInput = "(1,000000)" + vbLf + \_

"(39,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(99,000000)" + vbLf + \_

"(87,000000)"

End With

newanalysis.RouteOutput(newanalysis.Dialog.PrintPredictionsSpreadsheet).Visible = True

With newanalysis.Dialog

.RunUserDefinedInput = "(1,000000)" + vbLf + \_

"(40,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(0,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(1,000000)" + vbLf + \_

"(89,000000)" + vbLf + \_

"(69,000000)"

End With

newanalysis.RouteOutput(newanalysis.Dialog.PrintPredictionsSpreadsheet).Visible = True

End Sub