

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**«Віртуальний тренажер «Одношаровий персептрон»
для дистанційного курсу «Інтелектуальні
інформаційні системи»»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Шелехов І.В.

Студентка групи ІН.мдн-91С

Гузь Є.А.

СУМИ 2020

Сумський державний університет
(назва вузу)

Факультет ІЗДВН Кафедра Комп'ютерних наук
Спеціальність 122 «Комп'ютерні науки»

Затверджую:
зав. кафедри _____

“ _____ ” _____ 20__ р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ**

Гузь Євгенію Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Віртуальний тренажер «Одношаровий перцептрон» для дистанційного курсу «Інтелектуальні інформаційні системи»

затверджую наказом по інституту від “ _____ ” _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми та постановка задачі. 2) Нейроподібна структура типу одношаровий перцептрон. 3) Інформаційне та програмне забезпечення

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1	<i>Аналіз проблеми та постановка задачі</i>		
2	<i>Нейроподібна структура типу одношаровий перцептрон</i>		
3	<i>Інформаційне та програмне забезпечення</i>		
4	<i>Оформлення кваліфікаційної магістерської роботи</i>		

Студент – дипломник _____

(підпис)

Керівник проекту _____

(підпис)

РЕФЕРАТ

Записка: 50 стор., 8 рис., 1 додаток, 13 джерел.

Об'єкт дослідження — процес проектування та реалізації інформаційного та програмного забезпечення віртуального тренажеру

Мета роботи — розробка та програмна реалізація віртуального тренажеру

Методи дослідження — технологія проектування віртуальних тренажерів.

Результати — проведено розробку та програмну реалізацію віртуального тренажеру, що дозволяє набути практичних навичок з використання штучної нейроподібної структури – одношарового перцептрон, сформовано вхідний математичний опис віртуального тренажеру, сформовано математичну модель функціонування штучної нейроподібної структури типу одношаровий перцептрон, сформовано математичну модель навчання штучної нейроподібної структури типу одношаровий перцептрон, на основі сформованих математичних моделей розроблено та програмно реалізовано алгоритми функціонування і навчання штучної нейроподібної структури типу одношаровий перцептрон, розроблено і програмно реалізовано алгоритм роботи віртуального тренажеру, що відображає основні етапи функціонування і навчання штучної нейроподібної структури, перевірено працездатність розробленого віртуального тренажеру на задачі формування лінійного вирішального правила для двох класів з використанням штучної нейроподібної структури типу одношаровий перцептрон.

ВІРТУАЛЬНИЙ ТРЕНАЖЕР, E-LEARNING, ПЕРЦЕПТРОН, ВАГОВІ
КОЕФІЦІЄНТИ, НАВЧАННЯ

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Інфо-комунікаційні технології в освіті	6
1.2 Перспективи застосування віртуальних тренажерів в освіті	11
1.3 Постановка задачі	14
2 НЕЙРОПОДІБНА СТРУКТУРА ТИПУ ОДНОШАРОВИЙ ПЕРЦЕПТРОН.....	16
2.1 Перцептронний підхід	16
2.2 Математична модель	18
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.....	22
3.1 Формування вхідних даних.....	22
3.2 Алгоритм навчання перцептрону	22
3.3 Короткий опис програмної реалізації	25
3.4 Результати тестування	26
ВИСНОВКИ.....	34
СПИСОК ЛІТЕРАТУРИ.....	35
ДОДАТОК.....	36

ВСТУП

У сучасному світі все більшу роль набуває інформатизація освіти. Якщо не вдаватися в подробиці, інформатизація освіти це - впровадження в установи системи освіти інформаційних засобів, а також інформаційної продукції і педагогічних технологій, що базуються на цих засобах. Інформатизація освіти, як частина процесу інформатизації суспільства стало неминучим кроком у розвитку освіти, так як постійна потреба в розвитку виробництва, збору, зберігання, передачі та автоматизації інформаційних процесів зумовила поряд з традиційними інформаційними технологіями (газети, книги, журнали), розвиток нових інформаційних технологій . Нові інформаційні технології (або інфо-комунікаційні технології) - процеси, методи збору, пошуку, аналізу і способи обробки інформації за допомогою обчислювальних можливостей ЕОМ.[1-4]

В роботі розглядається задача проектування і реалізація за допомогою інфо-комунікаційних технологій одного з елементів навчального контенту дисципліни – віртуального тренажеру.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Інфо-комунікаційні технології в освіті

На початку становлення нових інформаційних технологій їх високий прикладної потенціал розглядався в різних галузях суспільства, в тому числі і в освіті. І дійсно, на сучасному етапі розвитку інформатизації освіти виникають можливості більш різнобічного використання комп'ютерної техніки і засобів комунікаційного зв'язку, як в організації навчального процесу, так і в подальшій організаційно-методичній підтримки самостійної роботи в рамках традиційних технологій навчання. Використання основних теоретичних положень і принципів, які були розроблені раніше в теорії навчання і застосування можливостей сучасної комп'ютерної техніки дозволяють створити потужні програмно-дидактичні системи модернізації освітнього процесу, пошуку і підтримки принципово нових видів діяльності учня [1]. Однак, незважаючи на цілком певний потенціал розвитку нових інформаційних технологій, давні очікування переходу глобальних, національних і регіональних систем освіти на новий рівень, на жаль, часто не виправдовуються. Нові інформаційні технології мають потужні інструменти для роботи з текстовою, числовою і графічною інформацією, що становить основу освітнього середовища. Але все ж, незважаючи на ці переваги, прагнення підвищити якість освіти шляхом впровадження інноваційних перетворень на основі застосування нових інформаційних технологій поки залишається нереалізованим або слабо реалізованим.

У початковій освіті [2], незважаючи на всю корисність і переваги, що надаються новими інформаційними технологіями, основним фактором є соціалізація дітей через контакт з учителями та іншими учнями. Для більшості педагогів дана задача є фундаментальною, особливо затребуваною в більш розвинених країнах, де діти в родині з обома працюючими батьками часто залишаються наодинці з телевізором, з комп'ютером або мобільним телефоном з можливістю виходу в інтернет. Так само початкова школа вчить дітей

вчитися, і це ключова життєва навичка, без якої неможливо існувати в сучасну епоху без нових інформаційних технологій.

Однією з найвідоміших є програма OneLaptopPerChild [2] (OLPC) (Один ноутбук для дитини), ініційована Ніколасом Негропonte в 2005 році. Ідея була в тому, що діти можуть навчати один в одного шляхом спроб і помилок на надійному і дешевому освітньому інструменті. При цьому знання формуються учнем за допомогою власних дій без вчителя. Негропonte хотів, щоб діти вчилися в країнах, що розвиваються, за допомогою методики конструктивізму. Минуло вже більше десяти років і проект зазнав невдачі. Причина невдачі в тому, що не було вирішено жодного із стратегічних питань: не було чітких цілей, було дуже мало освітнього контенту в наданих комп'ютерах, не було ніякого прагнення пристосувати комп'ютери до існуючих освітніх систем і стандартів. Найяскравішим прикладом даної невдачі є закупівля спеціалізованих ноутбуків в Перу на суму 225 мільйонів доларів і як підсумок: низькі бали з математики і читання в учнів у віці 7-ми років в початковій школі.

Іншим яскравим прикладом є експеримент, що проводився в Індії, так звана програма HoleInTheWall (HITW) [2-3] (Розміщення в стіні). Його розробник Сугета Мітра починав з схожих припущень, що і Негропonte, але потім пішов трохи в іншому напрямку. Замість розміщення комп'ютерів в школах, він розмістив на різних дитячих майданчиках вбудований в цегляну стіну комп'ютер. В результаті діти освоїли комп'ютер для перегляду сторінок, ігор, створення документів і малювання картинок протягом декількох днів. Згодом це механізм був впроваджений використаний в інших країнах (Камбоджа, Південна Африка). Основним висновком з цього експерименту стало те, що процес навчання відбувається в групах. І хоча проект мав місце поза шкільної системи, експертами була проведена оцінка знань учасників даного експерименту: діти поліпшили свої знання з математики, але ніякого ефекту не спостерігалось для англійської мови і природних наук.

Таким чином, діти дуже легко навчаються роботі з комп'ютером без підтримки з боку вчителя. Однак на рівень освіченості дітей комп'ютер (або інші електронні засоби) впливають мало. Впровадження нових інформаційних технологій в початковій освіті ефективно при активній підготовці відповідних навчальних програм та відповідної підготовки викладацького складу.

Для середньої школи [3] картина невдач схожа з початковою школою. Однак в даному випадку існують стимули для виправлення цієї ситуації:

- комп'ютерна грамотність - в сучасному світі метою більшості систем освіти є комп'ютерно-грамотний випускник, який за допомогою комп'ютера може поліпшити або збагатити навчання і знання з інших предметів;
- політичних контекст - незалежно від докази зворотного, політики продовжують розглядати ІТ як найбільш найкоротший шлях до поліпшення освіти. Вони сподіваються, що використання технологій призведе до більш швидких результатів, ніж наполеглива праця над поліпшенням навчання і управління;
- підготовка вчителів - здатність вчителів ефективно використовувати комп'ютери в роботі з учнями, і даному випадку завдання стоїть не в зміні менталітету вчителів, а в тому, щоб дати їм впевненість в роботі з сучасним обладнанням в школі;
- створення матеріалу для навчальних програм - країни, які впроваджують комп'ютери в школи, заохочують надати матеріали, що відображають їх національну освітню програму;
- організація - впровадження комп'ютерів в школи вимагає фізичних та адміністративних мереж комунікацій.

Чуйне дотримання цих стимулів дозволить якісно поліпшити застосування нових інформаційних технологій в рамках середній школі.

У вищій освіті [1, 3] нові інформаційні технології на відміну від шкіл, грають настільки важливу роль, що вони стали сприйматися як належне. Все більший акцент зміщується в бік дистанційного навчання. Дистанційне

навчання це - організація навчального процесу із застосуванням зберігається в базі даних і використовуваної при підготовці освітніх програм інформації та забезпечують її обробку технологій, засобів, а також телекомунікаційних мереж, які забезпечують трансляцію по лініях зв'язку зазначеної інформації, взаємодія всіх учасників освітнього процесу. Виділяють наступні переваги дистанційного навчання:

- навчання в індивідуальному темпі;
- свобода і гнучкість навчання;
- доступність навчання для будь-якої людини;
- ефективне здійснення взаємозв'язку між викладачем і навчаються;
- технологічність навчального процесу;
- соціальне рівноправність;
- сприятливі умови для особистого самовираження учня.

Поряд із зазначеними достоїнствами існують деякі недоліки:

- відсутність вербальної взаємодії між які навчаються і викладачем;
- відсутність частини індивідуально-психологічних факторів, яка характерна для класичної освіти;
- необхідність доступу до джерел отримання матеріалів (інтернет або створення відповідної технічної бази);
- відсутність практичних занять і семінарів;
- недостатня організація ресурсу дистанційного навчання.

Прикладами дистанційного навчання є відкриті освітні ресурси і масові відкриті онлайн-курси.

Відкриті Освітні Ресурси [1] - з'явилися в 2001 році, після того як Массачусетський технологічний інститут (MIT) відкрив вільний доступ до матеріалів своїх навчальних курсів (OpenCourseWare). Відкриті освітні ресурси це - освітні або наукові ресурси, розміщені у вільному доступі або забезпечені ліцензією, роздільної вільне використання або переробку. В даний час існує декілька десятків вільних ліцензій (наприклад, BSD, GNU, GPL і

т.д.), які широко використовуються для комп'ютерних програм і технічної документації в різних країнах. Відкриті освітні ресурси включають в себе повні курси, навчальні матеріали, модулі, підручники, відео, тести, програмне забезпечення, а також будь-які інші засоби, для надання доступу до знань.

Іншим прикладом дистанційного навчання є масові відкриті онлайн-курси [4]. З'явилися в результаті розвитку технологій дистанційної освіти і використанні цифрового контенту. Масові відкриті онлайн-курси спирається на наступні концепції: різноманітність підходів, підхід до навчання як до процесу формування мережі та прийняття рішень, навчання і пізнання як динамічний процес. Масові відкриті онлайн-курси - це електронні курси (навчально-методичні комплекси), що включають в себе відеолекції з субтитрами, текстові конспекти лекцій, домашні завдання, тести та підсумкові іспити. Авторами курсів є викладачі провідних університетів. Масові відкриті онлайн-курси спираються на активну участь і взаємодія студентів з викладачами і між собою. Після вивчення такого онлайн-курсу можливе отримання офіційного сертифіката. Також можна перерахувати наступні переваги масових відкритих онлайн-курсів:

- носять глобальний характер;
- виходять за рамки одного університету;
- усувають територіальні і тимчасові бар'єри;
- підвищують самостійність і мотивацію студентів в придбанні навичок, необхідних для професійної діяльності.

Як приклад масових відкритих онлайн-курсів можна привести сайт Stepic.org, однак найбільш найбільшим прикладом масових відкритих онлайн-курсів є платформа Coursera, що об'єднує понад 60-ти університетів по всьому світу. На рис. 1.1 зображена головна сторінка сайту.



Рисунок 1.1 – Сторінка з вибором курсів на платформі Coursera

У підсумку, нові інформаційні технології надійно зміцнилися в системі вищої освіти. Велике поширення набуває дистанційне навчання (ДН). Відповідно до розвитку і поширенням дистанційного навчання, в освіті з'являється нова тенденція, заснована на наступних напрямках:

- поширення доступного, дистанційної освіти, технологічною основою якого служать нові інформаційні технології і засоби телекомунікації;
- стандартизація наповнення і методології навчання, що вирішується шляхом повсюдного впровадження і поширення електронних форм подання та передачі матеріалу.

1.2 Перспективи застосування віртуальних тренажерів в освіті

Застосування сучасних засобів навчання служать до підвищення рівня підготовки кадрів. Розробка нових методів і засобів навчання вважається актуальним. В сучасних умовах поліпшення і підвищення ефективності та якості навчання вважається одним з важливих соціально-педагогічних вимог. З цієї причини необхідно створити ефективні засоби навчання і впровадити їх у практику. В ряду сучасних засобів навчання, що служать підвищенню ефективності, імітаційні віртуальні тренажери займають особливе місце. У практиці навчання їх використання служить поліпшенню і підвищенню

ефективності якості навчання. В процесі навчання застосування імітаційних віртуальних тренажерів дозволяє позитивно вирішити такі завдання [3, 4]:

- інформування (підручники, довідники, діапозитиви, діафільми, навчальні плакати, транспаранти, магнітні записи, кінофільми, відеозаписи, алгоритми, покажчики, інформаційно-технологічні карти, навчальні програми і тренажери);
- організація (алгоритми, покажчики, інформаційно-технологічні карти, навчальні програми і тренажери); управління (тести, навчальні програми і тренажери).

Смислове значення поняття «тренажер» (від англ. «train») означає «виховувати», «навчати», «вправляти». Тренажер є навчальним засобом, з його допомогою якого моделюються реальні умови майбутньої діяльності, формуються навички певної діяльності (наприклад, керування машиною (механізмом), робота на складному верстаті або навчання секретам військової техніки), навички перетворюються в кваліфікацію, а освітні кваліфікації удосконалюються. Тренажери призначені для формування в учнів практичних навичок в результаті спеціальної вправи, вони виконують такі завдання, як перевірка роботи на машині (механізмі), верстаті, електричного кола, електричному приладі, літаку, пароплаві тощо. [4]

Термін «віртуальний» перекладається з латинської мови, як «що відбувається, але не видимий, що не володіє формою». Поняття «реальність» («життя») означає «усвідомлена частина буття». «Віртуальна реальність» (інакше кажучи «віртуальне життя») означає «усвідомлена можливість» або «можливість усвідомлення». Поняття «віртуальний» в перекладі з англійської означає «що містить сутність деяких речей, які насправді не існують». У українській мові інтерпретують це поняття як «річ, що володіє реальною сутністю, але практично не існує». Віртуальність забезпечує «пожвавлення» в комп'ютері певних соціальних положень і безпосередню участь користувача комп'ютера в цьому процесі. У деяких джерелах дано таке визначення поняттю «віртуальне життя»: віртуальне життя - це технологія тривимірного взаємного

руху людини і комп'ютера, здійснювана за допомогою сукупних мультимедійно-оперативних засобів. Віртуальні технології - модель процесу створення, а також застосування методів, засобів впровадження в практику віртуальних образів і середовища для взаємного руху між ними або з ними. Тут потрібне розуміння сутності поняття «моделі». [1-4]

Імітація (від лат. «Imitation» - «наслідування») - комусь або чомусь наслідувати, уподібнювати, приймати, повторювати [3]. Діапазон застосування імітаційних віртуальних тренажерів дуже широкий включає в себе наступне:

- профорієнтація учнів;
- вхідне опитування;
- вивчення основних принципів рухів досліджуваних процесів або інструментів;
- формування розумових моторних і сенсорно-моторних навичок;
- освоєння сукупних навичок;
- освоєння навичок і кваліфікацій ефективного використання імітаційних віртуальних тренажерів;
- оцінка ступеня сформованості професійних навичок і кваліфікації учнів на підставі використання імітаційних віртуальних тренажерів.

Перевага імітаційних віртуальних технологій полягає в тому, що існує можливість введення з їх допомогою в певні процеси, що утворюються всередині і за допомогою комп'ютера. Саме з цієї особливості імітаційні віртуальні тренажери мають особливе дидактичне значення. З їх допомогою учні отримують можливість повністю освоїти сутність процесів, що необхідно для формування знань, навичок і придбання кваліфікації. В ефективності процесу формування практичних навичок і перетворення їх в кваліфікацію «спеціальні комп'ютерні засоби»- імітаційні віртуальні тренажери - займають особливе місце. В даний час віртуальні види тренажерів поділяються на такі типи [4]:

- 1) імітаційний;

- 2) умовний;
- 3) висвітлюючий;
- 4) обмежений.

Імітаційні віртуальні тренажери за короткий час і в реальних умовах забезпечують професійну підготовку фахівця. Перевага імітаційних віртуальних тренажерів полягають в тому, що в процесі їх використання майбутній фахівець [3-4]:

- розкриває можливості нових зв'язків;
- створює нові способи дистанційного керування;
- з психологічної точки зору швидко пристосовується до віртуального життя;
- в процесі вправ має можливість контролювати хід виробничого процесу;
- освоює кваліфікацію і вміння розподіляти свої фізичні і психічні сили;
- в процесі освоєння нових знань, навичок кваліфікації досягає повної свободи;
- визнаючи свої помилки, усуває їх через повторення імітаційного віртуального положення.

Ефективність імітаційного віртуального тренажера залежить від його технічних можливостей і методів підготовки, учнів до роботи на тренажері. Важливі вимоги, що пред'являються до методів їх використання, полягає в тому, щоб набуті за допомогою імітаційного віртуального тренажера навички переходили в ефективність розв'язання практичних задач.

1.3 Постановка задачі

Метою роботи є розробка та програмна реалізація віртуального тренажера, що дозволяє набуті практичних навичок з використання штучної нейроподібної структури – одношарового перцептроні. При цьому основними завданнями, що необхідно виконати для досягнення поставленої мети, є:

1. Сформулювати вхідний математичний опис віртуального тренажера.

2. Сформувати математичну модель функціонування штучної нейроподібної структури типу одношаровий перцептрон.
3. Сформувати математичну модель навчання штучної нейроподібної структури типу одношаровий перцептрон.
4. Визначити критерій ефективності навчання штучної нейроподібної структури.
5. Розробити і програмно реалізувати алгоритм роботи віртуального тренажеру, що відображає основні етапи функціонування і навчання штучної нейроподібної структури.
6. Перевірити працездатність розробленого віртуального тренажеру.

2 НЕЙРОПОДІБНА СТРУКТУРА ТИПУ ОДНОШАРОВИЙ ПЕРЦЕПТРОН

2.1 Перцептронний підхід

У 1957 році американський фізіолог Ф. Розенблатт зробив спробу технічно реалізувати фізіологічну модель сприйняття. Він виходив з припущення, що сприйняття здійснюється мережею нейронів. Згідно з поширеною і найбільш простою моделлю нейрона (моделі Мак-Калока - Пітса), нейрон - це нервова клітина, яка має кілька входів - дендритів і один вихід - аксон. Входи бувають або збуджуючі, або гальмуючі. Нейрон збуджується і посилає імпульси в тому випадку, якщо число сигналів, що прийшли по збудливим входах, перевершує число сигналів, що прийшли по гальмуючих входах нейрона [5].

Модель сприйняття складається з рецепторного шару, шару перетворюючих нейронів і шару реагуючих нейронів (рис. 2.1).

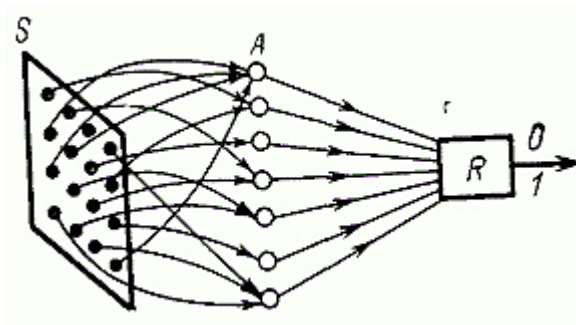


Рисунок 2.1 – Модель сприйняття

Зовнішнє роздратування сприймається рецепторами. Кожен рецептор пов'язаний з одним або декількома нейронами перетворюючого шару, при цьому кожен нейрон перетворюючого шару може бути пов'язаний з декількома рецепторами.

Виходи перетворюючих (асоціативних) нейронів в свою чергу з'єднуються з входами нейронів третього шару. Нейрони цього шару - реагуючі - теж мають кілька входів (дендритів) і один вихід (аксон), який

збуджується, якщо сумарна величина вхідних сигналів перевершує поріг спрацьовування. Але на відміну від нейронів другого шару, де підсумовуються сигнали з одним і тим же коефіцієнтом посилення (але, можливо, різними знаками), для реагуючих нейронів коефіцієнти підсумовування різні за величиною і, можливо, за знаком [5].

Кожен рецептор може бути в одному з двох станів: збудженому або незбудженому. Залежно від характеру зовнішнього роздратування в рецепторному шарі утворюється та чи інша множина імпульсів, яка, поширюючись по нервових шляхах, досягає шару перетворюючих нейронів. Тут утворюється букет множин імпульсів другого шару, яка надходить на входи реагуючих нейронів.

Сприйняття будь-якого об'єкта визначається збудженням відповідного нейрона третього шару, причому різними множинами імпульсів рецепторного шару може відповідати збудження одного і того ж реагуючого нейрона. Гіпотеза як раз і полягає в тому, що коефіцієнти посилення реагуючого нейрона підібрані так, щоб в разі, коли об'єкти належать до одного класу відповідні множини імпульсів збуджували б один і той же нейрон реагуючого шару. Наприклад, спостерігаючи який-небудь предмет в різних ракурсах, людина ототожнює побачене, оскільки кожен раз на різні зовнішні роздратування реагує один і той же нейрон, відповідальний за впізнавання цього предмета.

Серед величезного числа (порядку 10^{10}) нейронів людини, які забезпечують сприйняття, лише деяка частина зайнята сформованими вже поняттями, інша служить для утворення нових. Формування нового поняття, по суті, полягає у встановленні коефіцієнтів посилення реагуючих нейрона. Процес встановлення коефіцієнтів посилення реагуючих нейронів в схемі Розенблатта описується в термінах заохочення і покарання. [5-6]

Припустимо, що з'явилася множина імпульсів, яка відповідає раніше відомому поняттю. Якщо при цьому відповідний реагуючий нейрон не перейшов в збуджений стан (сигнал, що прийшов, не віднесено до даного

поняття), то реагуючий нейрон «штрафується»: коефіцієнти підсилення тих його входів, за якими проходив імпульс, збільшуються на одиницю. Якщо нейрон правильно реагував на імпульси, що прийшли, то коефіцієнти підсилення не змінюються. Якщо ж виявиться, що деякий набір сигналів буде помилково віднесений до даного поняття, то нейрон теж «штрафується»: в цьому випадку коефіцієнти посилення тих входів, за якими прийшов імпульс, зменшуються на одиницю. Така модель сприйняття проста і може бути реалізована на однорідних елементах - порогових елементах.

2.2 Математична модель

Пороговим називається елемент, який має n входів: x_1, x_2, x_n і один вихід y , причому сигнал на виході y може приймати тільки два значення, 0 і 1, і пов'язаний з входами співвідношенням [5]

$$y = \begin{cases} 0, & \text{якщо } \sum_{i=1}^n w_i x_i \geq w_0, \\ 1, & \text{якщо } \sum_{i=1}^n w_i x_i < w_0, \end{cases} \quad (2.1)$$

де w_1, w_2, w_n - коефіцієнти посилення сигналів, а w_0 - величина порога спрацьовування елемента.

Моделлю перетворюючого нейрона може служити пороговий елемент, у якого $w_0 = \pm 1$, а моделлю реагуючого нейрона служить пороговий елемент, у якого коефіцієнти w_1, w_2, w_n - певні числа, значення яких налаштовуються.

Технічну модель зорового аналізатора Розенблатт назвав перцептроном (від слова «перцепція» - сприйняття). Перший, рецепторний шар S моделі Розенблатта складався з набору 400 фотоелементів, які утворювали поле рецепторів (20x20). Сигнал з фотоелементів надходив на входи порогових елементів-нейронів перетворюючого шару (елементів A). Всього в моделі Розенблатта було 512 елементів. Кожен елемент A мав 10 входів, які випадковим чином були з'єднані з рецепторами - фотоелементами. Половина

входів вважалася гальмуючими і мала коефіцієнт посилення -1 , а інша половина - збуджуючими з коефіцієнтом посилення $+1$. Поріг спрацьовування нейрона приймався рівним нулю. Нарешті, сигнали з виходів елементів A надходили на входи реагуючого нейрона - елемента R (рис. 2.1) [5-7].

Персептрон призначався для роботи в двох режимах: в режимі навчання і в режимі експлуатації. У режимі навчання у персептрона за описаним вище принципом вироблялися величини коефіцієнтів w_1, w_2, \dots, w_n реагуючих нейронів. В ході експлуатації персептрон класифікував пред'явлені йому ситуації: якщо збуджувався j -й реагуючий елемент і не збуджувалися інші - R елементи, то ситуація відносилася до j -му класу. Ця машина, що отримала назву «Марк-1», була створена для експериментальної перевірки здатності персептрона утворювати поняття. Поява машини, здатної утворювати нові поняття, виявилось надзвичайно цікавим не тільки для фізіологів, а й для представників інших галузей знань і в першу чергу для математиків. Адже як тільки стала зрозуміла схема майбутньої експериментальної установки, персептрон перестав бути лише технічним аналогом фізіологічного феномена, він став математичною моделлю процесу сприйняття.

Визначення закону навчання новому поняттю - вироблення коефіцієнтів посилення кожного з елементів - означало завдання алгоритму, що вирішував деяке формальне завдання.

Ідея персептрона була здійснена технічно в машині «Марк-1». Однак для моделювання процесу сприйняття зовсім немає необхідності будувати спеціальну машину. Схема персептрона легко реалізується на ЕОМ, досить написати відповідну програму. Надалі вивчення феномена сприйняття пішло по шляху моделювання машин, що здатні навчатися, засобами ЕОМ, тобто по шляху створення програм, що навчаються.

Звернемося до математичної моделі функціонування персептрона [5]:

1. У рецепторному полі утворюється сигнал, відповідний до зовнішнього подразника, який подається деяким вектором x . Розенблатт зазначає, що кожне нервово закінчення передає досить простий сигнал - або

посилає імпульс, або не посилає його. Це означає, що вектор x бінарний, тобто його координати можуть приймати тільки два значення: 0 і 1.

2. Множина імпульсів поширюється до тих пір, поки за допомогою нейронів другого шару не буде перетворена в нову множину імпульсів (бінарний вектор x перетворюється в бінарний вектор y). Розенблатт уточнює характер перетворень $y=f(x)$:

а) перетворення здійснюється пороговими елементами;

б) входи перетворюючих порогових елементів з'єднані з рецепторами випадково.

3. Вважається, що персептрон відносить вхідний вектор до j -го поняття, якщо збуджується j -й реагуючий нейрон і не збуджуються інші реагуючі нейрони. Формально це означає, що для вектора $y=(y_1, y_2, \dots, y_m)$ виконується система нерівностей:

$$\sum_{i=1}^m w_{i,j} y_i \geq 0,$$

$$\sum_{i=1}^m w_{i,k} y_i < 0, \text{ для всіх } k \neq j.$$

У цих нерівностях $w_{i,k}$ - коефіцієнти посилення k -го реагуючого нейрона.

4. Формування понять в схемі Розенблатта зводиться до утворення коефіцієнтів (ваг) кожного з елементів R . Процедура побудови ваг елементів така.

Нехай до даного моменту існують деякі ваги елементів R і $w_{1,j}, w_{2,j}, \dots, w_{n,j}$ ваги j -того елемента. У момент часу t для класифікації на вхід персептрона надходить сигнал, що описується вектором $x(t)$. Вектор $x(t)$ може або відповідати поняттю j , або не відповідати йому. Розглянемо обидва ці випадки.

Випадок перший. Вектор $x(t)$ відповідає поняттю j . Тоді правильної реакцією елемента j на сигнал $x(t)$ має бути збудження, тобто має виконуватися нерівність

$$\sum_{i=1}^m w_{i,j} y_i \geq 0.$$

Якщо ваги елемента j забезпечують правильну реакцію на вектор $x(t)$, то вони не змінюються. Якщо ж ваги не забезпечують правильної реакції елемента j , тобто вони такі, що

$$\sum_{i=1}^m w_{i,j} y_i < 0,$$

то ваги елемента j змінюються за правилом

$$w_{i,j}(t) = w_{i,j}(t-1) + y_i.$$

Випадок другий. Вектор $x(t)$ не відповідає поняттю . Тоді елемент j не повинен збуджуватися, тобто має виконатися нерівність

$$\sum_{i=1}^m w_{i,j} y_i < 0.$$

Якщо ваги елемента j забезпечують правильну реакцію цього елемента на вектор $x(t)$, то вони не змінюються. Якщо ж ваги елемента j не забезпечують правильної реакції, тобто

$$\sum_{i=1}^m w_{i,j} y_i \geq 0,$$

то ваги, змінюються за правилом

$$w_{i,j}(t) = w_{i,j}(t-1) - y_i.$$

При навчанні аналогічно змінюються ваги всіх елементів R перцептрона.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Формування вхідних даних

При формуванні вхідних даних для віртуального тренажера враховувалась навчальна спрямованість розробки. Навчальна матриця складалася з невеликого числа реалізацій. При цьому реалізації представляли собою точки на площині, які описувалися двома координатами (x_1, x_2) . Основною умовою при формуванні реалізацій кожного класу було те, що вони повинні представляти собою опуклу множину точок, що не перетинаються з множиною реалізацій іншого класу. У цьому випадку, згідно з основним наслідком про роздільність двох опуклих множин теореми Гана-Банаха, такі множини є лінійнорозділимі або роздільними за допомогою гіперплощини. Для точок на площині, які використовуються як вхідні дані для розроблюваного тренажера, така гіперплощина вироджується в пряму. Навчальна матриця, яка використовувалася в роботі, наведена в таблиці 3.1

Таблиця 3.1 - Вхідні дані

	Клас X_1^o					Клас X_2^o				
	1	2	3	4	5	1	2	3	4	5
x_1	10	10	10	11	9	10	11	9	11	9
x_2	18	16	15	14	12	9	11	10	10	10

Навчальна матриця в табл. 3.1 характеризується потужністю словника ознак $N = 2$, потужністю алфавіту класів $m = 2$ та обсягом кожного класу $n = 5$.

3.2 Алгоритм навчання перцептронів

Перцептрон повинен вирішувати задачу класифікації за бінарним вхідним сигналам. Набір вхідних сигналів будемо позначати N -мірним вектором. Всі елементи вектора є булеві змінними (змінними, що приймають значення "True" або "False"). Однак іноді корисно оперувати числовими

значеннями. Будемо вважати, що значення "False" відповідає числове значення 0, а значення "True" відповідає 1. [7-10]

Персептроном будемо називати пристрій, що обчислює таку систему функцій:

$$y = \left[\sum_{i=1}^N w_i x_i \geq w_0 \right]$$

де w_i - ваги персептрона, w_0 - поріг, x_i - значення вхідних сигналів, дужки означають перехід від булевих (логічних) значень до числовим значенням за правилами, викладеним вище.

Навчання персептрона складається в підстроювання вагових коефіцієнтів. Нехай є набір пар векторів, $\{x_j, y_j\}$, так звана навчальна вибірка. Будемо називати нейронну мережу навченої на такій навчальній вибірці, якщо при подачі на входи мережі кожного вектора x_j на виходах щоразу виходить відповідний вектор y_j .

Запропонований Ф. Розенблатом метод навчання [8-9] полягає в ітераційному налаштуванню матриці ваг, послідовно зменшує помилку в вихідних векторах. Алгоритм включає кілька кроків:

Крок 0 Початкові значення $w_i(t=0)$ ваг задаються випадковим чином.

Крок 1 Мережі пред'являється вхідний образ x_j , в результаті формується вихідний образ t_j

Крок 2 Обчислюється вектор помилки $\delta_j = y_j - t_j$, що робиться мережею на виході. Подальша ідея полягає в тому, що зміна вектора вагових коефіцієнтів в області малих помилок повинно бути пропорційно помилці на виході і дорівнює нулю, якщо помилка дорівнює нулю.

Крок 3 Вектор ваг модифікується за такою формулою:

$$w_i(t+1) = w_i(t) + \eta x_j \delta_j$$

де η - швидкість навчання.

Крок 4 Кроки 1-3 повторюються для всіх навчальних векторів. Один цикл, коли на вхід мережі подається вся вибірка, називається епохою. Навчання завершується після закінчення декількох епох: а) коли ітерації зійдуться, тобто вектор ваг перестає змінюватися, або б) коли сума абсолютного значення помилок по всіх вхідних векторах стане менше деякого мінімального значення.

Пояснимо даний алгоритм більш детально. Подаємо на вхід перцептрона такий вектор, для якого вже відомо правильний вихід. Якщо вихідний сигнал перцептрона збігається з правильною відповіддю, то ніяких дій робити не треба. У разі помилки, необхідно навчити перцептрон правильно вирішувати даний приклад. Помилки можуть бути двох типів. Розглянемо кожен з них.

Перший тип помилки: на виході перцептрона - 0, а правильна відповідь - 1. Для того щоб перцептрон видавав правильну відповідь, необхідно, щоб сума $\sum_{i=1}^N w_i x_i$ стала більше. Оскільки змінні приймають значення 0 або 1, збільшення суми може бути досягнуто за рахунок збільшення вагів w_i . Однак немає сенсу збільшувати ваги при змінних, які дорівнюють нулю. Таким чином, слід збільшити ваги при тих змінних x_i , які дорівнюють 1.

Перше правило. Якщо на виході перцептрона отримано 0, а правильну відповідь дорівнює 1, то необхідно збільшити ваги зв'язків між одночасно активними нейронами. При цьому вихідний перцептрон вважається активним.

Другий тип помилки: на виході перцептрона - 1, а правильна відповідь дорівнює нулю. Для навчання правильного рішення даного прикладу слід зменшити суму $\sum_{i=1}^N w_i x_i$. Отже, необхідно зменшити ваги зв'язків при тих змінних, які дорівнюють 1 (оскільки немає сенсу зменшувати ваги зв'язків при рівних нулю змінних). Необхідно також провести цю процедуру для всіх активних нейронів попередніх шарів. В результаті отримуємо друге правило.

Друге правило. Якщо на виході персептрона отримана одиниця, а правильну відповідь дорівнює нулю, то необхідно зменшити ваги зв'язків між одночасно активними нейронами.

Таким чином, процедура навчання зводиться до послідовного перебору всіх прикладів навчальної множини із застосуванням правил навчання для помилково вирішених прикладів. Якщо після чергового циклу пред'явлення всіх прикладів виявиться, що всі вони вирішені правильно, то процедура навчання завершується.

Нерозглянутими залишилися два питання. Перший - про збіжність процедури навчання. Другий - на скільки потрібно збільшувати (зменшувати) ваги зв'язків при застосуванні правил навчання.

Відповідь на перше питання дають наступні теореми [10].

Теорема про збіжність персептрона. Якщо існує вектор параметрів, при якому персептрон правильно вирішує всі приклади навчальної вибірки, то при навчанні персептрона за алгоритмом, що подано вище, рішення буде знайдено за кінцеве число кроків.

Теорема про "зациклення" персептрона. Якщо не існує вектор параметрів, при якому персептрон правильно вирішує всі приклади навчальної вибірки, то при навчанні персептрона по даному правилу через кінцеве число кроків вектор ваг почне повторюватися.

Таким чином, дані теореми стверджують, що, запустивши процедуру навчання персептрона, через кінцевий час або отримає навчився персептрон, або відповідь, що даний персептрон поставленому завданню навчитися не може.

3.3 Короткий опис програмної реалізації

Додаток створено для web середовища на мовах Javascript, HTML і CSS [11-13]. Проект складається з таких файлів:

- index.html – основний файл, він зберігає всю структуру тренажера, блоки з завданнями і вхідними даними;

- script.js – зберігає всю логіку роботи і всі функції;
- style.css – файл каскадних таблиць стилів, використовується для візуального оформлення;
- jquery.min.js – бібліотека, яка використовується в основних функціях програми (табл. 3.2).

Таблиця 3.2 – Функції програми

Назва функції	Опис
remove_class(e)	Функція, яка прибирає підсвічування з полів введення, які користувач відредагував, але ще не перевірів.
goto_2()	Перехід до першого завдання
goto_tf(e)	Перевірка правильності вибору стану виконання умови.
goto_sel(e)	Перевірка вибору найближчого негативного елемента.
goto_inp(e)	Перевірка правильності введених даних в текстове поле.

Програмний код реалізації наведено в додатку

3.4 Результати тестування

Для перевірки працездатності тренажера оптимізуємо параметри персептрона Розенблата (рис. 3.1):

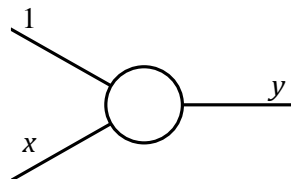


Рисунок 3.1 – Модель персептрона

При цьому:

$$\begin{aligned}x &\in R^N \text{ або } x \in \{-1, 1\}^N; \\ y &\in R \text{ або } y \in \{-1, 1\}\end{aligned}$$

Таку модель використовується для вирішення задачі класифікації для двох класів. Вона є ідентичною завдання:

$$y \in \{0, 1\}.$$

Будемо розглядати випадок:

$$x \in R^N; y \in \{-1, 1\}.$$

Функціонування персептрона описується наступною залежністю:

$$y = \text{sign}(\mathbf{w}^T \mathbf{x} - \tau) = f(\mathbf{x}, \mathbf{w}), \quad (3.1)$$

де τ - деякий поріг, \mathbf{w} - вектор вагових коефіцієнтів персептрону.

Вхідна навчальна матриця представлена в табл. 3.1. Перша сцена, яка з'являється при запуску тренажера відображає вхідні дані (рис. 3.2).

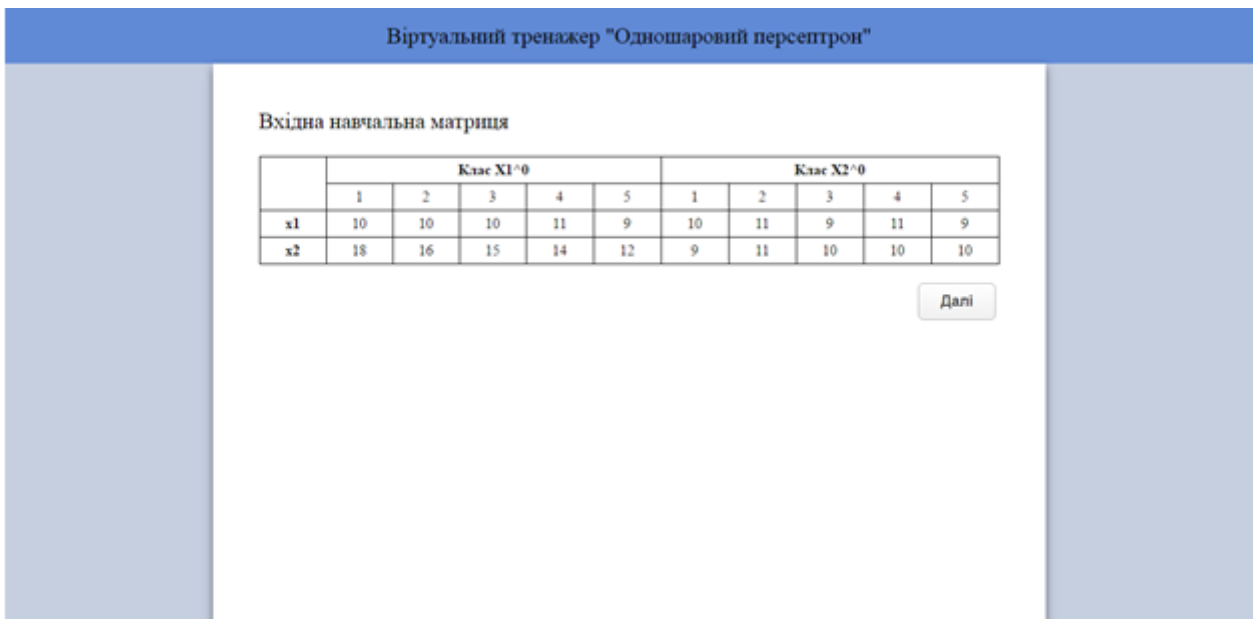


Рисунок 3.2 - Вхідні дані

Збільшуємо розмірність простору ознак (табл. 3.3):

Таблиця 3.3 - Результати першого кроку навчання

	Клас X_1^0					Клас X_2^0				
	1	2	3	4	5	1	2	3	4	5
x_1	10	10	10	11	9	10	11	9	11	9
x_2	18	16	15	14	12	9	11	10	10	10
x_3	1	1	1	1	1	1	1	1	1	1

Далі студент натискає на кнопку «Далі» і з'являється перше завдання – збільшити розмірність матриці (рис. 3.3).

Віртуальний тренажер "Одношаровий перцептрон"

Вхідна навчальна матриця

	Клас X1^0					Клас X2^0				
	1	2	3	4	5	1	2	3	4	5
x1	10	10	10	11	9	10	11	9	11	9
x2	18	16	15	14	12	9	11	10	10	10

Збільшуємо розмірність простору ознак

	Клас X1^0					Клас X2^0				
	1	2	3	4	5	1	2	3	4	5
x1	10	10	10	11	9	10	11	9	11	9
x2	18	16	15	14	12	9	11	10	10	10
x3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 3.3 - Перше завдання

Студенту потрібно заповнити пропуски в таблиці і натиснути кнопку «Далі» для перевірки відповідей. Правильні відповіді підсвічуються зеленим, а неправильні - червоним (рис. 3.4)

Віртуальний тренажер "Одношаровий перцептрон"

Вхідна навчальна матриця

	Клас X1^0					Клас X2^0				
	1	2	3	4	5	1	2	3	4	5
x1	10	10	10	11	9	10	11	9	11	9
x2	18	16	15	14	12	9	11	10	10	10

Збільшуємо розмірність простору ознак

	Клас X1^0					Клас X2^0				
	1	2	3	4	5	1	2	3	4	5
x1	10	10	10	11	9	10	11	9	11	9
x2	18	16	15	14	12	9	11	10	10	10
x3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 3.4 - Приклад підсвічування відповідей

Далі формуємо безліч $z_{ij} = x_{ij} x_{3j}$ (табл. 3.4)

Таблиця 3.4 - Результати другого кроку навчання персептрона

j	Клас X_1^o					Клас X_2^o				
	1	2	3	4	5	6	7	8	9	10
z_1	10	10	10	11	9	-10	-11	-9	-11	-9
z_2	18	16	15	14	12	-9	-11	-10	-10	-10
z_3	1	1	1	1	1	-1	-1	-1	-1	-1

Для відображення завдання, введення множини \mathbf{z} і перевірки правильності відповідей використовується форма аналогічна формі на рис. 3.2-3.3.

Далі вибираємо певний елемент $\mathbf{z}^{(1)}$ як початкове наближення для $\tilde{\mathbf{w}}$:

$$\tilde{\mathbf{w}} = \mathbf{z}^{(1)} = \begin{pmatrix} 10 \\ 18 \\ 1 \end{pmatrix}.$$

Перевіряємо виконання умови $\mathbf{z}^{(j)T} \tilde{\mathbf{w}} > 0$ для всіх векторів навчальної вибірки:

$$\mathbf{z}^T \tilde{\mathbf{w}} = (425 \ 389 \ 371 \ 363 \ 307 \ -263 \ -309 \ -271 \ -291 \ -271)$$

Проводимо модифікацію вагових коефіцієнтів $\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(l)}$, де $\varphi = 1$, $l = 6$.

$$\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(6)} = \begin{pmatrix} 10 \\ 18 \\ 1 \end{pmatrix} + \begin{pmatrix} -10 \\ -9 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \\ 0 \end{pmatrix}.$$

Перевіряємо виконання умови $\mathbf{z}^{(j)T} \tilde{\mathbf{w}} > 0$ для всіх векторів навчальної вибірки:

$$\mathbf{z}^T \tilde{\mathbf{w}} = (162 \ 144 \ 135 \ 126 \ 108 \ -81 \ -99 \ -90 \ -90 \ -90)$$

Проводимо модифікацію вагових коефіцієнтів, $\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(l)}$, де $\varphi = 1$, $l = 6$.

$$\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(6)} = \begin{pmatrix} 0 \\ 9 \\ 0 \end{pmatrix} + \begin{pmatrix} -10 \\ -9 \\ -1 \end{pmatrix} = \begin{pmatrix} -10 \\ 0 \\ -1 \end{pmatrix}.$$

Перевіряємо виконання умови $\mathbf{z}^{(j)T} \tilde{\mathbf{w}} > 0$ для всіх векторів навчальної вибірки:

$$\mathbf{z}^T \tilde{\mathbf{w}} = (-101 \ -101 \ -101 \ -111 \ -91 \ 101 \ 111 \ 91 \ 111 \ 91)$$

Проводимо модифікацію вагових коефіцієнтів, $\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(l)}$, де $\varphi = 1$, $l = 1$.

$$\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(1)} = \begin{pmatrix} -10 \\ 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 10 \\ 18 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 18 \\ 0 \end{pmatrix}.$$

Перевіряємо виконання умови $\mathbf{z}^{(j)T} \tilde{\mathbf{w}} > 0$ для всіх векторів навчальної вибірки:

$$\mathbf{z}^T \tilde{\mathbf{w}} = (324 \ 288 \ 270 \ 252 \ 216 \ -162 \ -198 \ -180 \ -180 \ -180)$$

Проводимо модифікацію вагових коефіцієнтів, $\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(l)}$, де $\varphi = 1$, $l = 6$.

$$\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}^{(6)} = \begin{pmatrix} 0 \\ 18 \\ 0 \end{pmatrix} + \begin{pmatrix} -10 \\ -9 \\ -1 \end{pmatrix} = \begin{pmatrix} -10 \\ 9 \\ -1 \end{pmatrix}.$$

Перевіряємо виконання умови $\mathbf{z}^{(j)T} \tilde{\mathbf{w}} > 0$ для всіх векторів навчальної вибірки:

$$\mathbf{z}^T \tilde{\mathbf{w}} = (61 \ 43 \ 34 \ 15 \ 17 \ 20 \ 12 \ 1 \ 21 \ 1)$$

Для відображення даних ітерацій в тренажері використовується набір подібних елементів (рис. 3.5)

$\mathbf{w} = \mathbf{z}_{\text{in}}^{\sim} = \begin{pmatrix} 18 \\ 1 \end{pmatrix}.$

Перевіряємо виконання умови $\tilde{\mathbf{w}}^T \mathbf{z}_{\text{in}}^{(j)} > 0$ для всіх векторів навчальної вибірки

$\tilde{\mathbf{w}}^T \mathbf{z} = (\text{425} \ \text{389} \ \text{371} \ \text{363} \ \text{307} \ \text{-263} \ \text{-309} \ \text{-271} \ \text{-291} \ \text{-271})$

Умова виконується:

Перша компонента, що менше нуля:

Проводимо модифікацію вагових коефіцієнтів

$\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \varphi \mathbf{z}_{\text{in}}^{(6)} = \begin{pmatrix} 10 \\ 18 \\ 1 \end{pmatrix} + \begin{pmatrix} -10 \\ -9 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \\ 0 \end{pmatrix}.$

Рисунок 3.5 - Відображення шостої ітерації алгоритму

Після завершення виконання всіх завдань тренажер виводить лінійне вирішальне правило в графічній формі. (Рис. 3.5)

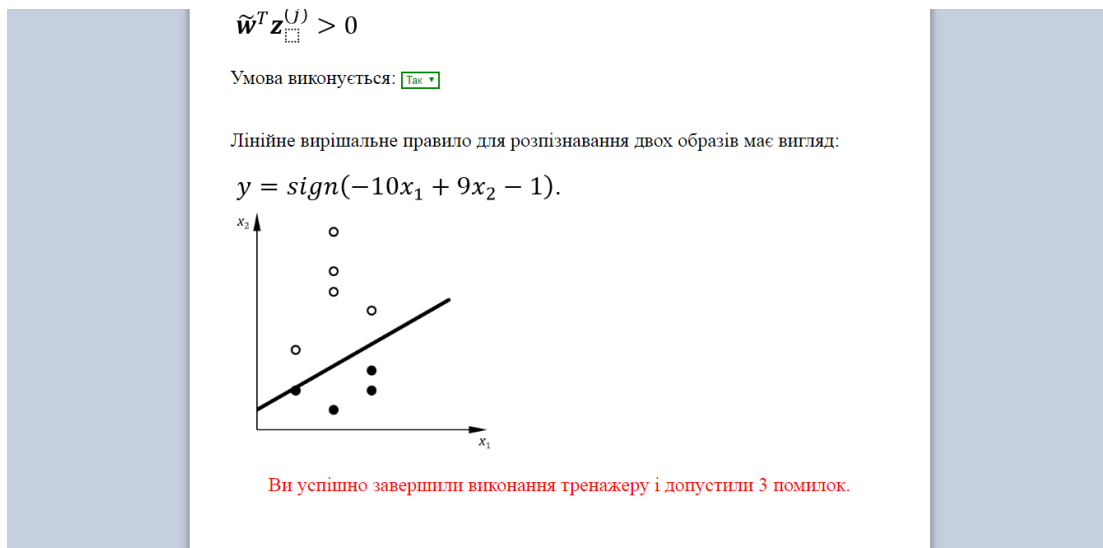


Рисунок 3.6 - Відображення результатів роботи тренажера

Крім того відображається інформація про кількість допущених помилок в ході проходження тренажера.

ВИСНОВКИ

В кваліфікаційній магістерській роботі проведено розробку та програмну реалізацію віртуального тренажеру, що дозволяє набути практичних навичок з використання штучної нейроподібної структури – одношарового перцептрон. При цьому виконано такі основні завдання:

1. Сформовано вхідний математичний опис віртуального тренажеру.
2. Сформовано математичну модель функціонування штучної нейроподібної структури типу одношаровий перцептрон.
3. Сформовано математичну модель навчання штучної нейроподібної структури типу одношаровий перцептрон.
4. На основі сформованих математичних моделей розроблено та програмно реалізовано алгоритми функціонування і навчання штучної нейроподібної структури типу одношаровий перцептрон.
5. Розроблено і програмно реалізовано алгоритм роботи віртуального тренажеру, що відображає основні етапи функціонування і навчання штучної нейроподібної структури.
6. Перевірено працездатність розробленого віртуального тренажеру на задачі формування лінійного вирішального правила для двох класів з використанням штучної нейроподібної структури типу одношаровий перцептрон.

СПИСОК ЛІТЕРАТУРИ

1. Breen Paul. Developing Educators for The Digital Age.– University of Westminster Press, 2018. – 220 p.
2. Crawford Emily. Online Teaching Tools: 3 Manuscripts: Google Classroom, Google Apps, Teaching with Zoom.– Springer, 2020. – 436 p.
3. Elena Dell'Aquila et al. Educational Games for SoftSkills Training in Digital Environments: New Perspectives. – Springer, 2017. – 178 p.
4. Liu D., Dede C., Huang R., Richards J. (eds.) Virtual, Augmented, and Mixed Realities in Education .– Springer, 2017. – 250 p.
5. Jyh-Horng Jeng, Jer-Guang Hsieh. Pathways to Machine Learning and Soft Computing. – EHGBooks, 2018. – 397 p.
6. Khan G.M. Evolution of Artificial Neural Development. In Search of Learning Genes. – New York: Springer, 2018. – 142 p.
7. McKeon Richard. Neural Networks for Electronics Hobbyists. –Apress, 2018. – 139 p.
8. Tiumentsev Y., Egorchev M. Neural Network Modeling and Identification of Dynamical Systems. – Academic Press, 2019. – 324 p.
9. Purkait N. Hands-On Neural Networks with Keras. – Packt, 2019. – 462 p.
10. Omelianenko I. Hands-on Neuroevolution with Python. – Packt, 2019. – 368 p.
11. McCaffrey J. Neural Networks with JavaScript Succinctly. –Morrisville (NC): Syncfusion, 2019. – 162 p.
12. Alves Claudia, Bach John. Javascript For Beginners: The Ultimate Guide to Understand JavaScript Code and Its Fundamentals.– 2nd edition. – Independently published, 2020. – 1189 p.
13. Du Preez O.J. JavaScript for Gurus: Use JavaScript programming features, techniques and modules to solve everyday problems. – New Delhi: BPB Publication, 2020. – 126 p.

ДОДАТОК

```

$( document ).ready(function() {
    $("#z_1").show();
    /*$("#input").each(function() {
        $(this).val($(this).attr("otvet"));
    });*/
});
var count_error = 0;
function remove_class(e) {
    $(e).removeClass("true");
    $(e).removeClass("false");
}
function goto_2() {
    $("#z_2").show();
    $("#z_1 button").hide();
}
function goto_tf(e) {
    if(e == 21) {
        var tr = 1;
    } else {
        var tr = 0;
    }
    if($("#z_" + e + " select :selected").val() == tr) {
        $("#z_" + e + " select").addClass("true");
        $("#z_" + e + " select").removeClass("false");
        $("#z_" + (e+1)).show();
        $("#z_" + e + " button").hide();
    } else {
        $("#z_" + e + " select").removeClass("true");
        $("#z_" + e + " select").addClass("false");
        count_error++;
    }
    if(e == 21) {
        $("#z_22").append('<p class="end">Ви успішно завершили виконання тренажеру і допустили '+count_error+' помилок.</p>');
    }
}
function goto_sel(e) {
    if(e == 14) {
        var tr = 1;
    } else {
        var tr = 6;
    }
    if($("#z_" + e + " select :selected").val() == tr) {
        $("#z_" + e + " select").addClass("true");
        $("#z_" + e + " select").removeClass("false");
        $("#z_" + (e+1)).show();
        $("#z_" + e + " button").hide();
    } else {
        $("#z_" + e + " select").removeClass("true");
        $("#z_" + e + " select").addClass("false");
    }
}

```

```

        count_error++;
    }
}
function goto_inp(e){
    var success = true;
    $("#z_" + e + " input").each(function(){
        if( (String($(this).attr("otvet")).replace("-", "-")) ==
(String($(this).val()).replace("-", "-"))){
            $(this).addClass("true");
            $(this).removeClass("false");
        }else{
            $(this).removeClass("true");
            $(this).addClass("false");
            success = false;
        }
    });
    if(success){
        $("#z_" + (e+1)).show();
        $("#z_" + e + " button").hide();
    }else{
        count_error++;
    }
}

```

Код программы на css:

```

html, body{
    min-height: 100%;
    margin: 0;
}
h2{
    font-size: 16pt;
    text-align: center;
}
#header{
    background-color: #618AD6;
    height: 60px;
    font-size: 18pt;
    text-align: center;
    line-height: 60px;
}
#wrapper{
    min-height: 100%;
    background-color: #C6CFE0;
}
#container{
    margin: auto;
    min-height: 100%;
    width: 900px;
    background-color: white;
    box-shadow: 0 5px 15px rgba(0,0,0,0.5);
}
#reshenie{
    margin: auto;
    width: 800px;
}

```

```

padding-bottom: 50px;
}
#reshenie div{
padding-top: 25px;
display: none;
font-size: 18pt;
}
.table {
width: 100%;
background: black;
border-spacing: 1px;
}
.table td {
background: white;
padding: 5px;
text-align: center;
}
.table_input{
width: 30px;
font-size: 12pt;
text-align: center;
}
.div_input{
width: 40px;
font-size: 12pt;
text-align: center;
}
.false{
border: 2px solid;
color: red;
border-color: red;
}
.true{
border: 2px solid;
color: green;
border-color: green;
}
.end{
color: red;
text-align: center;
}
button {
margin-left: 89%;
margin-top: 20px;
-moz-box-shadow:inset 0px 1px 0px 0px #ffffff;
-webkit-box-shadow:inset 0px 1px 0px 0px #ffffff;
box-shadow:inset 0px 1px 0px 0px #ffffff;
background:-webkit-gradient(linear, left top, left bottom,
color-stop(0.05, #ffffff), color-stop(1, #f0f0f0));
background:-moz-linear-gradient(top, #ffffff 5%, #f0f0f0
100%);
background:-webkit-linear-gradient(top, #ffffff 5%, #f0f0f0
100%);

```

```

background:-o-linear-gradient(top, #ffffff 5%, #f0f0f0 100%);
background:-ms-linear-gradient(top, #ffffff 5%, #f0f0f0
100%);
background:linear-gradient(to bottom, #ffffff 5%, #f0f0f0
100%);
filter:progid:DXImageTransform.Microsoft.gradient(startColors
tr='#ffffff', endColorstr='#f0f0f0',GradientType=0);
background-color:#ffffff;
-moz-border-radius:6px;
-webkit-border-radius:6px;
border-radius:6px;
border:1px solid #b8b8b8;
display:inline-block;
cursor:pointer;
color:#4d4d4d;
font-family:Arial;
font-size:16px;
font-weight:bold;
padding:10px 24px;
text-decoration:none;
text-shadow:0px 1px 0px #ffffff;
}
button:hover {
background:-webkit-gradient(linear, left top, left bottom,
color-stop(0.05, #f0f0f0), color-stop(1, #ffffff));
background:-moz-linear-gradient(top, #f0f0f0 5%, #ffffff
100%);
background:-webkit-linear-gradient(top, #f0f0f0 5%, #ffffff
100%);
background:-o-linear-gradient(top, #f0f0f0 5%, #ffffff 100%);
background:-ms-linear-gradient(top, #f0f0f0 5%, #ffffff
100%);
background:linear-gradient(to bottom, #f0f0f0 5%, #ffffff
100%);
filter:progid:DXImageTransform.Microsoft.gradient(startColors
tr='#f0f0f0', endColorstr='#ffffff',GradientType=0);
background-color:#f0f0f0;
}
button:active {
position:relative;
top:1px;
}

```

Код программы на index.html:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<link href="css/style.css" rel="stylesheet">
<script src="js/jquery.min.js"></script>
<script src="js/script.js"></script>
<title>Тренажер</title>
</head>

```



```

<body>
  <div id="wrapper">
    <div id="header">
      <span>Віртуальний тренажер "Одношаровий персептрон"</span>
    </div>
    <div id="container">
      <div id="reshenie">
        <div id="z_1">
          <p>Вхідна навчальна матриця</p>
          <table class="table">
            <tr><td rowspan=2></td><td colspan=5><b>Клас X1^0</b></td><td colspan=5><b>Клас X2^0</b></td></tr>
            <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
            <tr><td><b>x1</b></td><td>10</td><td>10</td><td>10</td><td>11</td><td>9</td><td>10</td><td>11</td><td>9</td><td>11</td><td>9</td></tr>
            <tr><td><b>x2</b></td><td>18</td><td>16</td><td>15</td><td>14</td><td>12</td><td>9</td><td>11</td><td>10</td><td>10</td><td>10</td></tr>
          </table>
          <button onclick="goto_2 ()">Далі</button>
        </div>
        <div id="z_2">
          <p>Збільшуємо розмірність простору ознак</p>
          <table class="table">
            <tr><td rowspan=2></td><td colspan=5><b>Клас X1^0</b></td><td colspan=5><b>Клас X2^0</b></td></tr>
            <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
            <tr><td><b>x1</b></td><td>10</td><td>10</td><td>10</td><td>11</td><td>9</td><td>10</td><td>11</td><td>9</td><td>11</td><td>9</td></tr>
            <tr><td><b>x2</b></td><td>18</td><td>16</td><td>15</td><td>14</td><td>12</td><td>9</td><td>11</td><td>10</td><td>10</td><td>10</td></tr>
            <tr><td><b>x3</b></td><td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
            <td><input class="table_input" oninput="remove_class(this)" type="text" otvet="1"></td>
          </table>
        </div>
      </div>
    </div>
  </div>

```

```

    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
</tr>
</table>
<button onclick="goto_inp(2)">Далі</button>
</div>
<div id="z_3">
<p>Формуємо множини </p>
<table class="table">
<tr><td rowspan=2><b>j</b></td><td colspan=5><b>Клас
X1^0</b></td><td colspan=5><b>Клас X2^0</b></td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>1</
td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td><b>z1</b></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="10"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="10"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="10"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="11"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="9"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-11"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-9"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-11"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-9"></td>
</tr>
<tr><td><b>z2</b></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="18"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="16"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="15"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="14"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="12"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-9"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-11"></td>

```

```

    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
</tr>
<tr><td><b>z3</b></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
    <td><input class="table_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
</tr>
</table>
<button onclick="goto_inp(3)">Далі</button>
</div>
<div id="z_4">
    <p>Вибираємо деякий елемент  як
початкове наближення для </p>
    <br>
    <p>Перевіряємо виконання умови  для всіх векторів навчальної вибірки</p>
     = (
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="425">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="389">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="371">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="363">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="307">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-263">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-309">

```

```

    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-271">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-291">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-271">
)
<br><button onclick="goto_inp(4)">Дали</button>
</div>
<div id="z_5">
<br>
Умова виконується:
<select oninput="remove_class(this)">
<option value="1">Так</option>
<option value="0">Hi</option>
</select><br>
<button onclick="goto_tf(5)">Дали</button>
</div>
<div id="z_6">
Перша компонента, що менше нуля:
<select oninput="remove_class(this)">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
</select><br>
<button onclick="goto_sel(6)">Дали</button>
</div>
<div id="z_7">
<p>Проводимо модифікацію вагових коефіцієнтів</p>
<table>
<tr>
<td rowspan=3></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
<td rowspan=3></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="0"></td>
<td rowspan=3></td>
</tr>
<tr>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-9"></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="9"></td>
</tr>
<tr>

```

```

    <td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
    <td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="0"></td>
</tr>
</table>
<button onclick="goto_inp(7)">Далі</button>
</div>
<div id="z_8">
<p>Перевіряємо виконання умови  для всіх векторів навчальної вибірки</p>
     = (
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="162">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="144">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="135">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="126">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="108">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-81">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-99">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-90">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-90">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-90">
    )
<br><button onclick="goto_inp(8)">Далі</button>
</div>
<div id="z_9">
<br><br>
Умова виконується:
<select oninput="remove_class(this)">
<option value="1">Так</option>
<option value="0">Ні</option>
</select><br>
<button onclick="goto_tf(9)">Далі</button>
</div>
<div id="z_10">
Перша компонента, що менше нуля:
<select oninput="remove_class(this)">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>

```

```

<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
</select><br>
<button onclick="goto_sel(10)">Далі</button>
</div>
<div id="z_11">
<p>Проводимо модифікацію вагових коефіцієнтів</p>
<table>
<tr>
<td rowspan=3></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
<td rowspan=3></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
<td rowspan=3></td>
</tr>
<tr>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-9"></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="0"></td>
</tr>
<tr>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
</tr>
</table>
<button onclick="goto_inp(11)">Далі</button>
</div>
<div id="z_12">
<p>Перевіряємо виконання умови  для всіх векторів навчальної вибірки</p>
 = (
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="-101">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="-101">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="-101">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="-111">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="-91">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="101">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="111">

```

```



```

```

    <td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="1"></td>
    <td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="0"></td>
</tr>
</table>
<button onclick="goto_inp(15)">Далі</button>
</div>
<div id="z_16">
<p>Перевіряємо виконання умови  для всіх векторів навчальної вибірки</p>
     = (
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="324">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="288">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="270">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="252">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="216">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-162">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-198">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-180">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-180">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="-180">
    )
<br><button onclick="goto_inp(16)">Далі</button>
</div>
<div id="z_17">
<br><br>
Умова виконується:
<select oninput="remove_class(this)">
<option value="1">Так</option>
<option value="0">Ні</option>
</select><br>
<button onclick="goto_tf(17)">Далі</button>
</div>
<div id="z_18">
Перша компонента, що менше нуля:
<select oninput="remove_class(this)">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>

```



```

<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
</select><br>
<button onclick="goto_sel(18)">Далі</button>
</div>
<div id="z_19">
<p>Проводимо модифікацію вагових коефіцієнтів</p>
<table>
<tr>
<td rowspan=3></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
<td rowspan=3></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-10"></td>
<td rowspan=3></td>
</tr>
<tr>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-9"></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="9"></td>
</tr>
<tr>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
<td><input class="div_input" oninput="remove_class(this)"
type="text" otvet="-1"></td>
</tr>
</table>
<button onclick="goto_inp(19)">Далі</button>
</div>
<div id="z_20">
<p>Перевіряємо виконання умови  для всіх векторів навчальної вибірки</p>
 = (
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="61">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="43">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="34">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="15">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="17">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="20">
<input class="div_input" oninput="remove_class(this)"
type="text" otvet="12">

```

```

    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="1">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="21">
    <input class="div_input" oninput="remove_class(this)"
type="text" otvet="1">
)
<br><button onclick="goto_inp(20)">Далі</button>
</div>
<div id="z_21">
<br><br>
Умова виконується:
<select oninput="remove_class(this)">
<option value="1">Так</option>
<option value="0">Ні</option>
</select><br>
<button onclick="goto_tf(21)">Далі</button>
</div>
<div id="z_22">
<p>Лінійне вирішальне правило для розпізнавання двох образів
має вигляд:</p>
<br>
<br>
</div>
</div>
</div>
</div>
</body>

```