

Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

МЕТОДИ ТА СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ

**Методичні рекомендації та
завдання для лабораторних робіт**

Для студентів спеціальностей:

«Комп'ютерні науки»,
«Програмна інженерія»,
«Комп'ютерна інженерія»

Чернівці
Чернівецький національний університет
2022

УДК 004.81, 004.85, 004.89, 004.93

Друкується за ухвалою Вченої ради Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича, протокол № 8 від 22 вересня 2022 р.

Методи та системи штучного інтелекту: Методичні вказівки та завдання для лабораторних робіт. Укл. В.Г.Фратавчан, Т.М.Фратавчан. – Чернівці: ЧНУ, 2022. – 38 с.

Навчальне видання містить завдання для лабораторних робіт та теоретичні відомості, необхідні для їх виконання, з дисципліни «Методи та системи штучного інтелекту».

Для студентів, що здобувають освіту в галузі знань „Інформаційні технології” та інших спеціальностей з поглибленим вивченням інформатики.

Укладачі:

Фратавчан Валерій Григорович, кандидат фізико-математичних наук, доцент кафедри математичних проблем управління і кібернетики,

Фратавчан Тоня Михайлівна, кандидат фізико-математичних наук, доцент кафедри математичного моделювання Чернівецького національного університету ім.Ю.Федьковича

ВСТУП

Дисципліна «Методи та системи штучного інтелекту» є складовою частиною якісної підготовки фахівців в області ІТ технологій і ознайомлює студентів з новітніми та сучасними підходами до розв'язування складних прикладних задач з інтелектуальним трактуванням.

Метою викладання навчальної дисципліни «Методи та системи штучного інтелекту» є

- ознайомлення студентів з новітніми та перспективними комп'ютерними технологіями;
- освоєння методів та алгоритмів штучного інтелекту;
- ознайомлення з методами еволюційного програмування.

Основними завданнями вивчення дисципліни «Методи та системи штучного інтелекту» є:

- ознайомлення з проблемами штучного інтелекту;
- вивчення послідовних та структурних методів розпізнавання образів;
- вивчення структур та математичних основ побудови нейронних мереж та їх використання в реальних задачах;
- вивчення принципів еволюційного програмування та застосування генетичних алгоритмів в складних задачах оптимізації;
- розуміння математичної побудови природних та формальних мов.

Згідно з вимогами освітньо-професійної програми студенти повинні знати :

- загальні задачі та математичні моделі штучного інтелекту;
- класифікацію систем розпізнавання образів;
- класифікацію алгоритмів розпізнавання образів;
- математичні моделі розпізнавання образів;
- основні алгоритми класифікації, навчання та розпізнавання образів;
- схему перцептрона;
- схеми основних нейронних мереж;
- алгоритми навчання та функціонування основних типів нейронних мереж;
- схему генетичного алгоритму;
- загальні підходи опису та аналізу формальних та природних мов.

вміти :

- створювати модель задачі розпізнавання образів;
- реалізувати етапи попередньої обробки та побудови ознакових векторів;
- реалізувати алгоритми навчання та класифікації в задачах розпізнавання образів;
- реалізувати схеми перцептронів та простих нейронних мереж в задачах розпізнавання образів;
- реалізувати генетичні операції та застосувати генетичний алгоритм в нетрадиційних задачах оптимізації.

Під час вивчення дисципліни формуються наступні компетентності (відповідно до галузевого стандарту):

Загальні компетентності:

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК6. Здатність вчитися й оволодівати сучасними знаннями.
- ЗК8. Здатність генерувати нові ідеї (креативність).
- ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.
- ЗК13. Здатність діяти на основі етичних міркувань.

Спеціальні (фахові, предметні) компетентності:

СК1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування.

СК2. Здатність до виявлення статистичних закономірностей недетермінованих явищ, застосування методів обчислювального інтелекту, зокрема статистичної, нейромережевої та нечіткої обробки даних, методів машинного навчання та генетичного програмування тощо.

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.

СК11. Здатність до інтелектуального аналізу даних на основі методів обчислювального інтелекту включно з великими та погано структурованими даними, їхньої оперативної обробки та візуалізації результатів аналізу в процесі розв'язування прикладних задач.

СК17. Здатність класифікувати задачі, створювати та реалізовувати алгоритми синтезу оптимального керування процесів у кіберфізичних, економічних та соціальних динамічних системах.

Нормативний зміст підготовки здобувачів вищої освіти, сформульований у термінах результатів навчання:

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР3. Використовувати знання закономірностей випадкових явищ, їх властивостей та операцій над ними, моделей випадкових процесів та сучасних програмних середовищ для розв'язування задач статистичної обробки даних і побудови прогнозних моделей.

ПР4. Використовувати методи обчислювального інтелекту, машинного навчання, нейромережевої та нечіткої обробки даних, генетичного та еволюційного програмування для розв'язання задач розпізнавання, прогнозування, класифікації, ідентифікації об'єктів керування тощо.

ПР12. Застосовувати методи та алгоритми обчислювального інтелекту та інтелектуального аналізу даних в задачах класифікації, прогнозування, кластерного аналізу, пошуку асоціативних правил з використанням програмних інструментів підтримки багатовимірного аналізу даних на основі технологій DataMining, TextMining, WebMining.

Навчальна тема № 1: Проблеми штучного інтелекту. Класифікація задач. Основні математичні моделі.

Зміст теми:

1. Основні положення теорії штучного інтелекту.
2. Основні класи задач: розпізнавання образів, доведення теорем, розв'язування задач, побудова ігрової стратегії, розуміння людської мови, оптимізація в складних умовах.
3. Математичні моделі задач штучного інтелекту.

Теоретичні відомості.

1. Основні положення теорії штучного інтелекту.

Інтелект – здатність людини сприймати інформацію від зовнішнього середовища, оцінити ситуацію та приймати адекватні рішення.

Системою штучного інтелекту називається людино-машинна система, яка використовується для автоматизації процесів творчої діяльності людини в різних сферах, зокрема прийнятті рішень.

Термін «штучний інтелект» з'явився в 1956 році (семінар у Дортмунді). Теорія штучного інтелекту як наука розділена на дві складові: нейрокібернетика та кібернетика «чорної скрині».

Нейрокібернетика започаткована роботами Мак-Калока (1956 р.) та Розенблата (1962 р.), за якими була запропонована модель *перцептрона*. Особливий розвиток нейрокібернетики спостерігається в 80 –х роках.

Напрямок «чорної скриньки» будується на принципі, що структура інтелектуальної системи неважлива, головне щоб ІС реагувала на ситуації так само як людина. Цей напрямок був орієнтований на автоматизований пошук алгоритмів рішення інтелектуальних задач. Завдяки цього напрямку з'явилися мови штучного інтелекту (МакКарті, ЛІСП) та фреймові моделі (Мінський).

2. Основні класи задач штучного інтелекту.

До основних класів задач штучного інтелекту відносять:

- представлення знань, маніпуляції знаннями та створення експертних систем;
- спілкування, комунікації «людина - комп'ютер»;
- розпізнавання образів;
- навчання та самонавчання;
- планування дій, пошук розв'язків задач;
- нейронні мережі;
- самоорганізація, методи евристичної самоорганізації;
- еволюційне моделювання та генетичний алгоритм;

- автоматизація конструювання та проектування.

3. Математичні моделі задач штучного інтелекту.

В загальному математичному розумінні, інтелектуальна задача задається у формі:

$$P = \langle S, C, D, M, L, W \rangle$$

S – множина можливих ситуацій;

C - множина керувань для кожної ситуації;

D – множина завад;

M – множина переходів від однієї ситуації у іншу;

L – множина бажаних ситуацій;

W - множина небажаних ситуацій.

Навчальна тема № 2: Розпізнавання образів. Основні поняття. Системи РО. Математичні моделі.

Зміст теми:

1. Класифікація та розпізнавання. Поняття образу.
2. Класифікація методів. Основні поняття.
3. Класифікація систем розпізнавання образів.

Теоретичні відомості.

1. Класифікація та розпізнавання. Поняття образу.

Розпізнаванням називають процес розбиття загальної сукупності об'єктів на групи (класи) зі спільними визначеними параметрами.

Класифікацією (ідентифікацією) називають процес визначення класу, до якого відноситься об'єкт їх заданими параметрами.

Ознака – числова, предикатна, структурна, описова, топологічна характеристика, яку можна використовувати для формального опису об'єкта.

Простір ознак – загальна множина всіх можливих значень ознак.

Образ – опис одного об'єкта в термінах (за допомогою) ознак;

- точка у просторі ознак.

Клас- підмножина образів, які характеризуються спільними або близькими за значенням ознаками і займають у просторі ознак компактно обмежену область.

Вектор ознак - сукупність ознак, за допомогою яких можна відрізнити об'єкти різних класів.

2. Класифікація методів. Основні поняття теорії розпізнавання.

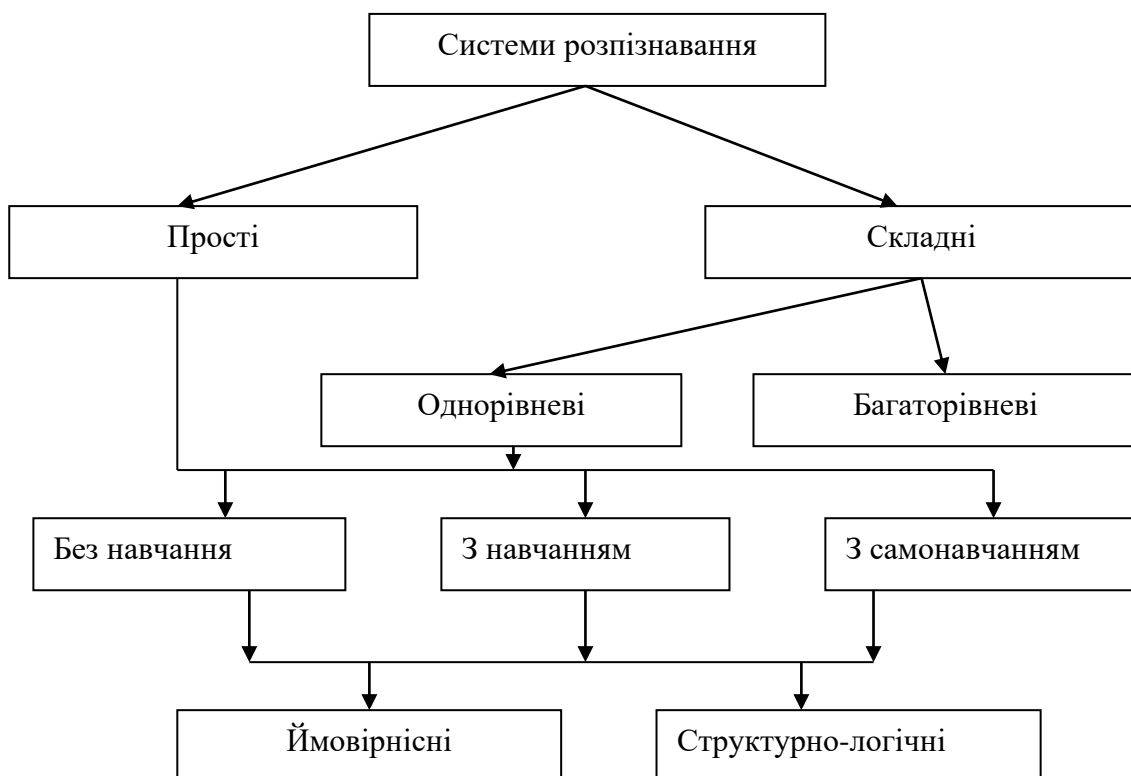
Система розпізнавання це апаратно-програмний комплекс для введення інформації, проведення процедур розпізнавання та класифікації. Процес створення системи розпізнавання складається з декількох етапів:

- виявлення числових, предикатних, структурних тощо ознак для опису об'єктів;
- побудова простору ознак;
- проведення попередню процедуру класифікації, аналіз розміщення класів у просторі ознак;
- модифікація, доповнення простору ознак та мінімізація його розміру;
- розробка процедур для створення формальних образів;
- розробка процедур розпізнавання;

Для проведення процедур розпізнавання та класифікації використовуються різні методи та алгоритмічні підходи. В залежності від типу використаних ознак методи розпізнавання діляться на дві загальні групи. **Паралельні методи** об'єднують алгоритми, в яких використовуються числові та предикатні ознаки. До **послідовних або структурних методів** відносять системи, де ідентифікація відбувається за описовими, топологічними та структурними ознаками. Паралельні методи, в свою чергу, можна розділити на декілька напрямків, в залежності від типу алгоритмів. Тут можна відмітити такі групи, як використання міри, порівняння з еталонами, роздільні функції.

3. Класифікація систем розпізнавання образів.

Системи розпізнавання, як апаратно-програмні комплекси, можна класифікувати наступною схемою:



Лабораторна робота № 1: Створення середовища для попередньої обробки образів та формування ознакових векторів в системах розпізнавання.

У візуальному середовищі програмування розробити програмний комплекс для побудови абсолютних та нормованих векторів ознак розпізнавання стандартизованих графічних зображень (цифр, букв, логотипів).

Середовище повинно виконувати наступні функції:

- Послідовне введення графічних зображень з графічних файлів. Розміри зображень – змінні, але образ займає весь прямокутник зображення і не потребує додаткової локалізації у графічному компоненті. Зображення чорно-білі і не потребують додаткової соляризації та фільтрації;
- Відображення образу у графічному компоненті;
- Проведення сегментації за варіантом. Схема сегментації відображається на зображенні;
- Підрахунок точок образу на фоні зображення у кожному секторі сегментації та обчислення абсолютного вектора ознак. Значення вектора ознак відобразити у відповідному компоненті;
- Проведення нормування ознакових векторів за вказаним варіантом та відображення нормованих значень.

Варіанти сегментації:

- 1) Задана кількість комірок прямокутної табличної схеми;
- 2) Задана кількість однакових за розміром вертикальних смуг;
- 3) Задана кількість однакових за розміром горизонтальних смуг;
- 4) Задана кількість кутових секторів з центром у геометричному центрі зображення. Нульовий кут – вправо. Порядок нумерації – проти годинникової стрілки.
- 5) Задана кількість кутових секторів з центром у верхньому лівому куті зображення. Порядок нумерації – за годинникової стрілки.
- 6) Задана кількість кутових секторів з центром у середині верхньої межі зображення. Порядок нумерації – проти годинникової стрілки.
- 7) Задана кількість кутових секторів з центром у верхньому лівому куті зображення. Порядок нумерації – за годинникової стрілки.
- 8) Задана кількість кутових секторів з центром у нижньому лівому куті зображення. Порядок нумерації – за годинникової стрілки.

Варіанти нормування:

- 1) За сумою 1 (діленням всіх елементів абсолютного вектора ознак на суму ознакових значень).
- 2) За модулем 1 (діленням всіх елементів абсолютного вектору ознак на максимальний елемент вектора).

Навчальна тема № 3: Системи РО детермінованого та ймовірнісного типу. Ознаки. Простори ознак, вектори ознак.

Зміст теми:

1. Поняття ознаки. Числові, предикатні та структурні ознаки.
2. Побудова числових векторів ознак.
3. Розбиття простору ознак на кластери. Поняття оцінки вектору ознак.
4. Алгоритми класифікації з використанням оцінок.

Теоретичні відомості.

1. Поняття ознаки. Числові, предикатні та структурні ознаки.

Ознаками називають числові, предикатні, структурні, описові, топологічні характеристики, які можна використовувати для формального опису об'єкта та його ідентифікації серед інших об'єктів.

Числові ознаки визначають деякі кількісні параметри об'єктів. Вони можуть мати чітко визначені значення (кількість лап у комах, кількість копит у ссавців тощо), і тоді ознаку відносять до детермінованого типу, або ознака може приймати випадкові значення в межах деякого діапазону (артеріальний тиск, температура тіла, вага, довжина тощо) і тоді ознаку вважають ймовірнісною.

Предикатні ознаки визначають наявність або відсутність у об'єктів дослідження деяких конкретних характеристик.

До описових ознак відносять вербальні або альтернативні ознаки, такі як колір, відносні розміри, середовище проживання, покриття тіла тощо.

Структурні ознаки визначають склад та взаємне розміщення структурних елементів досліджених об'єктів.

2. Побудова числових векторів ознак.

В загальному випадку деяку множину об'єктів можна описувати змішаною системою з числових, предикатних, структурних ознак. Якщо опис містить числові ознаки, то за допомогою предикатних, описових та структурних ознак відбувається попередня класифікація об'єктів з метою розбиття загальної множини класів на декілька груп. Числові ознаки використовуються для остаточної ідентифікації в межах групи.

Для сукупності числових ознак будуються відповідні ознакові простори, а опис одного об'єкту сукупністю числових характеристик називають числовим вектором ознак.

3. Розбиття простору ознак на кластери. Поняття оцінки вектору ознак.

При використанні числових векторів ознак кожний образ визначається точкою у просторі ознак. Оскільки в межах одного класу кожна ознака може отримати фіксовану кількість значень, або нескінчену кількість значень з деякого діапазону, можна вважати, що кожний клас займає у просторі ознак деяку обмежену область. Задача розпізнавання або кластеризації полягає у розбитті загального простору ознак на окремі області розміщення класів (кластери) та розробки процедур визначення параметрів кожного кластера. Існують декілька підходів опису або визначення класів у просторі ознак. В залежності від форми представлення класів, в системах розпізнавання застосовуються ті чи інші методи класифікації. В багатьох випадках в задачах кластеризації та класифікації використовується оцінка (міра) вектору ознак. Здебільшого, в таких задачах міра пов'язана з поняттям відстані. В теорії РО використовуються наступні терміни:

а) Відстань між точками.

Оскільки розглядаються точки в N-вимірному просторі, можна застосувати наступні відстані:

- Евклідова відстань.

$$d_1(X_1, X_2) = \left\{ \sum_k (x_k^1 - x_k^2)^2 \right\}^{1/2}$$

- Манхетенська відстань.

$$d_2(X_1, X_2) = \sum_k |x_k^1 - x_k^2|.$$

- Чебишевська відстань.

$$d_3(X_1, X_2) = \max_k |x_k^1 - x_k^2|.$$

б) Відстань від точки до класу.

$$D(P, X_0) = \inf \{d(P, M), P, M \in R^N, M \in X_0, X_0 \subset R^N\}.$$

в) Відстань між класами.

$$D(X_1, X_2) = \inf \{d(P, M), P, M \in R^N, P \in X_1, M \in X_2, X_1, X_2 \subset R^N\}.$$

4. Алгоритми класифікації з використанням оцінок.

При використанні векторів ознак для розпізнавання та класифікації використовуються наступні групи алгоритмів:

- локалізація класів областями прямокутної форми;
- локалізація класів областями сферичної (еліпсоїдальної) форми;
- обчислення мінімальних відстаней;

- використання статистичних (ймовірнісних) характеристик.

Лабораторна робота № 2: Створення системи розпізнавання для класифікації образів з детермінованими ознаками.

Використовуючи основні можливості першої лабораторної роботи, створити систему розпізнавання з нормованими векторами ознак числового типу за методом порівняння з еталоном. Система повинна забезпечити наступні дії:

- Ввести з графічних файлів еталонні образи для трьох класів та відобразити їх у відповідних графічних компонентах;
- Створення абсолютних та нормованих ознакових векторів для кожного еталону з відображенням абсолютних та нормованих еталонних значень;
- Ввести з графічного файлу зображення невідомого образу, відобразити образ у графічному компоненті;
- Обчислити абсолютний та нормований вектор ознак для невідомого образу та відобразити його ознакові значення;
- Обчислити міри відповідності невідомого образу до кожного еталону за варіантом норми;
- Провести класифікацію невідомого образу.

Варіанти норми:

- 1) Евклідова норма.
- 2) Норма Чебишева.
- 3) Манхетенська норма.

Лабораторна робота № 3: Створення адаптивних систем розпізнавання з побудовою кластерів методом статистичної обробки навчальних послідовностей.

Використовуючи основні можливості першої лабораторної роботи, створити систему розпізнавання з навчанням з нормованими векторами ознак числового типу. Система повинна забезпечити наступні дії:

- Ввести послідовно з графічних файлів образи навчальної послідовності для трьох класів та відобразити їх у відповідному графічному компоненті;
- Створення абсолютних та нормованих ознакових векторів для кожного зразка навчальної послідовності з відображенням абсолютних та нормованих ознакових значень;
- В режимі навчання з вчителем провести кластеризацію поточного навчального зразка;
- Провести поточну статистичну обробку кожного кластера за варіантом і відобразити його поточні статистичні параметри;
- Ввести з графічного файлу зображення невідомого образу, відобразити образ у графічному компоненті;
- Обчислити абсолютний та нормований вектор ознак для невідомого образу та відобразити його ознакові значення;

- Провести класифікацію невідомого образу за його вектором ознак та вибраним методом класифікації.

Варіанти кластер-аналізу:

- 1) Обчислення «геометричних центрів» кластерів за математичним сподіванням векторів ознак навчальної послідовності у просторі та Евклідової міри в декартовій системі координат;
- 2) Обчислення мінімальних та максимальних значень кожної компоненти кожного кластера у просторі ознак за зразками навчальних послідовностей та перевірки належності до кожного кластера.

Навчальна тема № 4: Специфічні математичні моделі в розпізнаванні образів. Метод відокремлюючих функцій. Ймовірнісний підхід.

Зміст теми:

1. Поняття відокремлюючої функції.
2. Алгоритм побудови відокремлюючих функцій лінійного типу.
3. Байєсівський підхід до РО. Загальний ймовірнісний підхід. Побудова ймовірнісних оцінок.

Теоретичні відомості.

1. Поняття відокремлюючої функції.

Нехай кожний образ задається вектором ознак $X = (x_1, \dots, x_n)$. Потрібно розпізнавати образи класів C_1, C_2, \dots, C_L , $C_i \cap C_j = \emptyset, i \neq j$.

$d(X) = C$ – розв’язальне правило, за яким образ X віднесений до класу C .

Означення: Відокремлююча функція – це деяка функція $g_i(x)$ класу C_i , така, що

$$d(x) = C_i, \Leftrightarrow g_i(x) \geq g_j(x), i = 1, \dots, L.$$

2. Алгоритм побудови відокремлюючих функцій лінійного типу.

Образ задається вектором ознак $X = (x_1, \dots, x_n)$. Його можна доповнити додатковою компонентою з одиничним значенням $\bar{X} = (1, x_1, \dots, x_n)$. Йому ставиться у відповідність вектор коефіцієнтів $\bar{W} = (w_0, w_1, \dots, w_n)$.

Відокремлюючу функцію для i -го класу задають у вигляді

$$g_i(X) = \bar{W}^T \cdot \bar{X} = w_0 + w_1 * x_1 + \dots + w_n * x_n.$$

У випадку двох класів C_1, C_2 розв'язувальне правило має вигляд:

$$d(X) = \begin{cases} 1, & g_1(X) \geq g_2(X), \\ 2, & g_1(X) < g_2(X). \end{cases}$$

Для всієї системи можна визначити загальну відокремлюючу функцію

$$g(X) = g_1(X) - g_2(X).$$

Тоді розв'язувальне правило має вигляд

$$d(X) = \begin{cases} 1, & g(X) \geq 0, \\ 2, & g(X) < 0 \end{cases} = -\frac{1}{2}(\text{sign}[g(X)] - 3).$$

3. Байєсівський підхід до РО. Загальний ймовірнісний підхід. Побудова ймовірнісних оцінок.

а) *Бейєсівський підхід.*

Нехай $P(A)$ - ймовірність події A ,

$P(A \cap B)$ - ймовірність сукупності подій A та B ,

$P(A/B)$ - ймовірність події A за умови, що відбулась подія B .

За формулою умовної ймовірності $P(A \cap B) = P(A) * P(B/A)$.

Нехай об'єкти мають описуватися вектором ознак X . Об'єкти належать до N класів. $P(i)$ - ймовірність класу i . $P(X/i)$ - ймовірність того, що у класі i існує екземпляр з вектором ознак X . Тоді ймовірність належності вектору ознак X_0 до класу k обчислюється за формулою:

$$P(k/X_0) = \frac{P(X_0/k)P(k)}{\sum_{i=1}^N P(X_0/i)P(i)}.$$

Розпізнаним вважається клас з максимальною умовною ймовірністю.

б) *Класичний ймовірнісний підхід.*

Нехай розпізнаються два класи Ω_1 і Ω_2 з використанням числової ймовірнісної ознаки X . Потрібно вибрати порогове значення X_0 і побудувати функцію класифікації

$$d(X) = \begin{cases} 1, & X \leq X_0, \\ 2, & X > X_0. \end{cases}$$

Для знаходження порогового значення використовується наступна методика. Під час обробки зразків навчальної послідовності кожного класу

обчислюються середньоарифметичні значення ознак m_1, m_2 та середньоквадратичні відхилення σ_1, σ_2 .

$$m = 1/N \sum_i X_i$$

$$\sigma^2 = 1/N \sum_i (X_i - m)^2$$

Припускаючи, що ознака є випадковою величиною з нормальним законом розподілу в межах кожного класу, формується **функція розподілу для кожного класу:**

$$f_i(x) = p(x/i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - m_i}{\sigma_i}\right)^2\right).$$

Вводиться поняття помилок класифікації. Імовірність **помилки першого роду**

$$Q_1 = \int_{x_0}^{\infty} f_1(x) dx - \text{ймовірність того що об'єкт з першого класу буде віднесено}$$

до другого класу.

Імовірність **помилки другого роду**

$$Q_2 = \int_{-\infty}^{x_0} f_2(x) dx - \text{ймовірність того що об'єкт з другого класу буде віднесено}$$

до першого класу.

Це дає можливість обчислити ймовірність вірної класифікації:

$$D_1 = 1 - Q_1,$$

$$D_2 = 1 - Q_2.$$

Вводиться матриця «штрафів» за прийняття рішень:

$$\|C\| = \begin{vmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{vmatrix}, \text{ де } c_{ij} - \text{штраф за віднесення образу з класу } i \text{ до класу } j.$$

Середні втрати за класифікацію обчислюються за формулою:

$$R = P(\Omega_1)c_{11}D_1 + P(\Omega_1)c_{12}Q_1 + P(\Omega_2)c_{22}D_2 + P(\Omega_2)c_{21}Q_2.$$

Відношення $\frac{f_1(x)}{f_2(x)}$ називають коефіцієнтом правдоподібності, а значення

$$\lambda_0 = \frac{P(\Omega_1)(c_{12} - c_{11})}{P(\Omega_2)(c_{21} - c_{22})} - \text{критична точка коефіцієнта правдоподібності.}$$

Порогове значення функції класифікації обчислюється за формулою:

$$x_0 = \frac{\sigma_2^2 m_1 - \sigma_1^2 m_2 \pm \sigma_1 \sigma_2 \sqrt{(m_2 - m_1)^2 + (\sigma_2^2 - \sigma_1^2) \ln \lambda_0 \frac{\sigma_2^2}{\sigma_1^2}}}{\sigma_2^2 - \sigma_1^2}$$

Частинні випадки:

а) $\sigma_1 = \sigma_2 = \sigma$.

$$x_0 = \frac{(m_1 + m_2)}{2} + \frac{\sigma^2}{(m_2 - m_1)} \ln \lambda_0.$$

б) $c_{11} = c_{22} = 0$; $c_{12} = c_{21}$; $P(\Omega_1) = P(\Omega_2)$.

$$x_0 = (m_1 + m_2) / 2.$$

Навчальна тема № 5: Біологічна та математична модель нейрону. Штучна нейронна мережа.

Зміст теми:

1. Структура нервової клітини. Функції елементів.
2. Математична модель нейрона. Вагові коефіцієнти. Передаточна функція.

Теоретичні відомості.

1. Структура нервової клітини. Функції елементів.

Нейрон (нервова клітка) складається з тіла клітини - соми (*soma*), і двох типів зовнішніх деревоподібних відгалужень: аксона (*axon*) і дендритів (*dendrites*). Тіло клітини вміщує ядро (*nucleus*), що містить інформацію про властивості нейрону, і плазму, яка продукує необхідні для нейрону матеріали. Нейрон отримує сигнали (імпульси) від інших нейронів через дендрити (приймачі) і передає сигнали, згенеровані тілом клітки, вздовж аксона (передавач), що наприкінці розгалужується на волокна (*strands*). На закінченнях волокон знаходяться синапси (*synapses*).

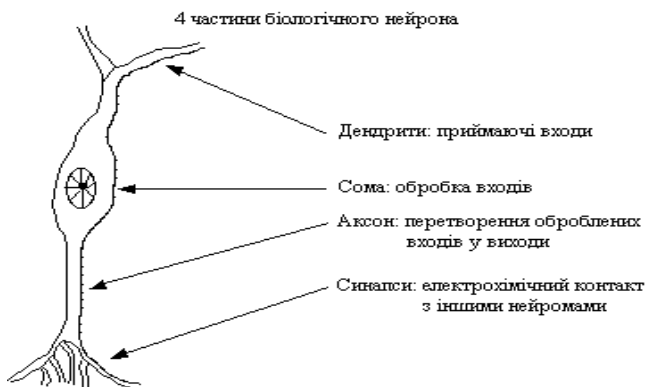


Рис. 1. Схема біологічного нейрона.

Синапс є функціональним вузлом між двома нейронами (волокно аксона одного нейрона і дендрит іншого). Коли імпульс досягає синаптичного закінчення, продукуються хімічні речовини, названі нейротрансмітерами. Нейротрансмітери проходять через синаптичну щілину, збуджуючи або загальмовуючи, у залежності від типу синапсу, здатність нейрона-приймача генерувати електричні імпульси. Результативність синапсу налаштовується минаючими через нього сигналами, тому синапси навчаються в залежності від активності процесів, у яких вони приймають участь. Нейрони взаємодіють за допомогою короткої серії імпульсів. Повідомлення передається за допомогою частотно-імпульсної модуляції.

Останні експериментальні дослідження доводять, що біологічні нейрони структурно складніші, ніж спрощене пояснення, наведене вище і значно складніші, ніж існуючі штучні нейрони, які є елементами сучасних штучних нейронних мереж. Оскільки нейрофізіологія надає науковцям розширене розуміння дії нейронів, а технологія обчислень постійно вдосконалюється, розробники мереж необмежений простір для вдосконалення моделей біологічного мозку.

2. Математична модель нейрона. Вагові коефіцієнти. Передаточна функція.

Базовий модуль нейронних мереж штучний нейрон моделює чотири основні функції природного нейрона (рис. 2).

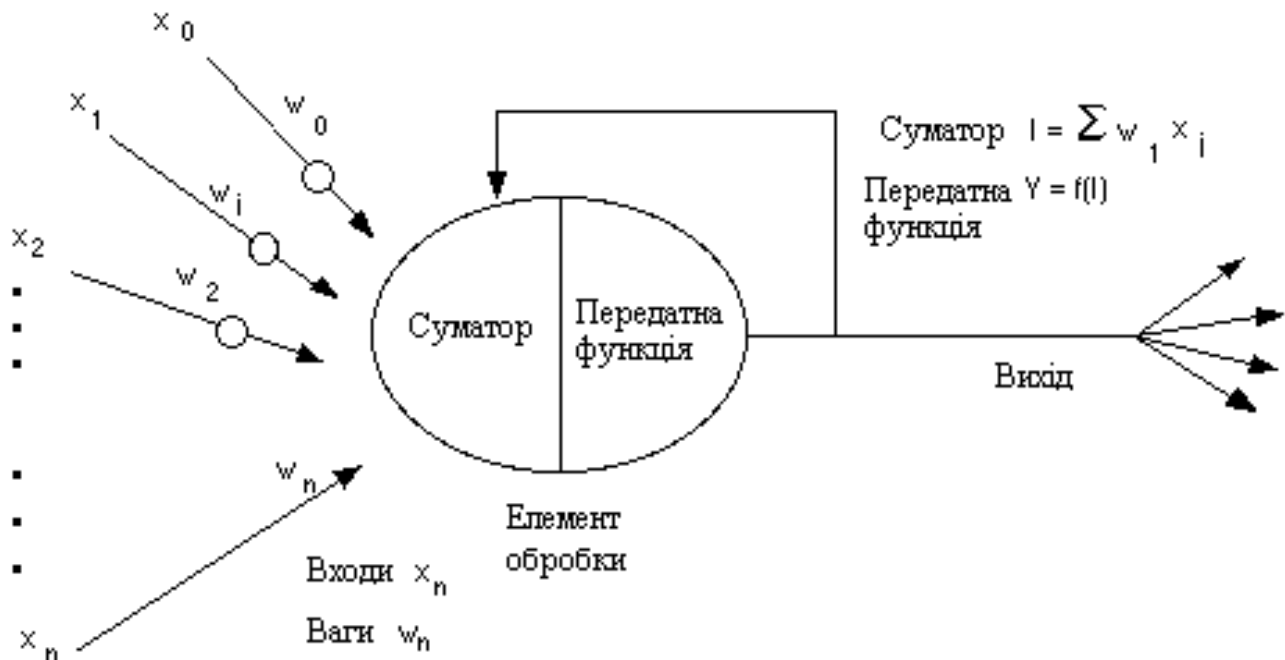


Рис. 2. Базовий штучний нейрон.

Вхідні сигнали x_n зважені ваговими коефіцієнтами з'єднання w_n додаються, проходять через передатну функцію, генерують результат і виводяться.

Навчальна тема № 6: Прості нейронні архітектури. Перцептрон Розенблата. Алгоритм навчання та класифікації.

Зміст теми:

1. Поняття перцептрону. Структура перцептрону Розенблата для двох класів.
2. Схема навчання. Оцінка функціональних можливостей.
3. Модель штучного нейрона як «елемента обробки».

Теоретичні відомості.

1. Перцептрон Розенблата

Першою моделлю нейромереж вважають перцептрон Розенблата. Теорія перцептронів є основою для багатьох типів штучних нейромереж прямого поширення і вони є класикою для вивчення.

Одношаровий перцептрон здатний розпізнавати найпростіші образи. Окремий нейрон обчислює зважену суму елементів вхідного сигналу, віднімає значення зсуву і пропускає результат через жорстку порогову функцію, вихід якої дорівнює +1 чи -1. В залежності від значення вихідного сигналу приймається рішення:

- +1 - вхідний сигнал належить класу А,
- -1 - вхідний сигнал належить класу В.

На рис. 1 показана схема нейронів, використовуваних в одношарових перцептронах, графік передатної функції і схема вирішальних областей, створених у багатовимірному просторі вхідних сигналів. Вирішальні області визначають, які вхідні образи будуть віднесені до класу А, які - до класу В. Перцептрон, що складається з одного нейрона, формує дві вирішальні області, розділені гіперплощиною. На рисунку показаний випадок, коли розмірність вихідного сигналу дорівнює 2. При цьому поділяюча поверхня уявляє собою пряму лінію на площині. Рівняння, що задає поділяючу пряму, залежить від значень синапсичних ваг і зсуву.

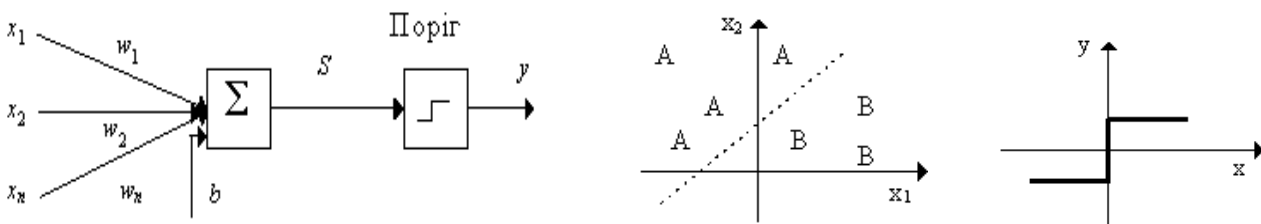


Рис. 1. Схема нейрона, графік передатної функції і поділяюча поверхня.

2. Алгоритм навчання одношарового перцептрону.

Алгоритм навчання одношарового перцептрона:

1. Ініціалізація синапсичних ваг і зсуву: синапсичні ваги приймають малі випадкові значення.
2. Пред'явлення мережі нового вхідного і бажаного вихідного сигналів: вхідний сигнал $x=(x_1, x_2, \dots, x_n)$ пред'являється нейрону разом з бажаним вихідним сигналом d .
3. Обчислення вихідного сигналу нейрона:
- 4.

$$y(t) = f\left(\sum_{i=1}^N w_i(t)x_i(t) - b\right)$$

5. Налаштування значень ваг:
- 6.

$$W_i(t+1) = w_i(t) + r[d(t) - y(t)]x_i(t), \quad i=1, \dots, N$$

$$d(t) = \begin{cases} +1, & \text{вихідний клас А} \\ -1, & \text{вихідний клас В} \end{cases}$$

Де $w_i(t)$ - вага зв'язку від i -го елемента вхідного сигналу до нейрона в момент часу t , r - швидкість навчання (менше 1); $d(t)$ - бажаний вихідний сигнал.

Якщо мережа приймає правильне рішення, синаптичні ваги не модифікуються.

7. Перехід до кроку 2.

Тип вхідних сигналів: бінарні чи аналогові (дійсні).

Розмірності входу і виходу обмежені при програмній реалізації тільки можливостями обчислювальної системи, на якій моделюється нейронна мережа, при апаратній реалізації - технологічними можливостями.

Області застосування: розпізнавання образів, класифікація.

Недоліки. Примітивні поділяючі поверхні (гіперплощини) дають можливість вирішувати лише найпростіші задачі розпізнавання.

Переваги. Програмні та апаратні реалізації моделі дуже прості. Простий і швидкий алгоритм навчання.

Модифікації. Багатошарові перцептрони дають можливість будувати більш складні поділяючі поверхні і тому більш поширені.

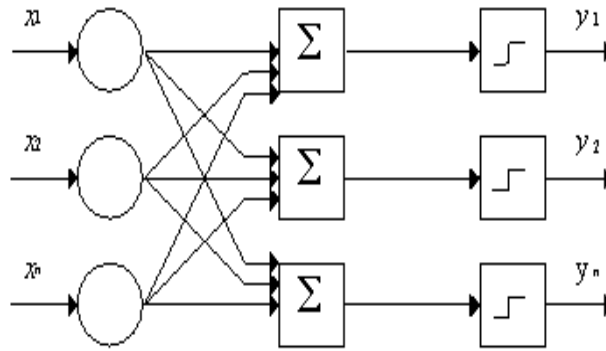


Рис. 2. Перцептрон із багатьма виходами.

3. Модель штучного нейрона як «елемента обробки».

У наявних на цей час пакетах програм штучні нейрони називаються "елементами обробки" і мають набагато більше можливостей, ніж простий штучний нейрон, описаний вище. На рис. 3 зображена детальна схема спрощеного штучного нейрону.

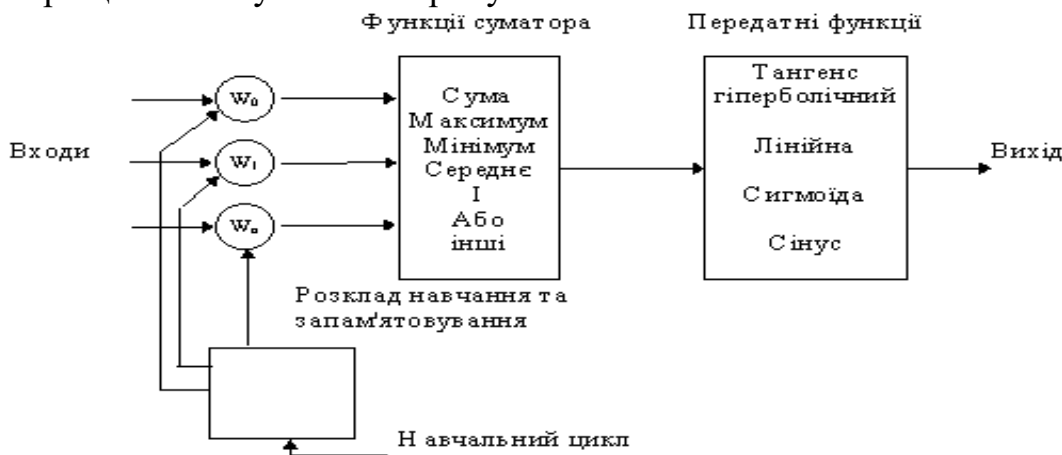


Рис. 3. Модель "елементу обробки".

Модифіковані входи передаються на функцію сумування, яка переважно тільки сумує добутки. Проте можна обрати багато різних операцій, такі як середнє, найбільше, найменше, *OR*, *AND*, тощо, які могли б виробляти деяку кількість різних значень. Окрім того, більшість комерційних програм дозволяють інженерам-програмістам створювати власні функції сумування за допомогою підпрограм, закодованих на мові високого рівня (C, C++, turborascal). Інколи функція сумування ускладнюється додаванням функції активації, яка дозволяє функції сумування оперувати в часі.

В будь-якому з цих випадків, вихід функції сумування надсилається у передатну функцію і скеровує весь ряд на дійсний вихід (0 або 1, -1 або 1, або яке-небудь інше число) за допомогою певного алгоритму. В існуючих нейромережах в якості передатних функцій можуть бути використані сигмоїда, синус,

гіперболічний тангенс та ін. Приклад того, як працює передатна функція показаний на рис. 4.



Рис. 4. Сигмоїдна передатна функція.

Після обробки сигналу, нейрон на виході має результат передатної функції, який надходить на входи інших нейронів або до зовнішнього з'єднання, як це передбачається структурою нейромережі.

Всі штучні нейромережі конструюються з базового формуючого блоку - штучного нейрону. Існуючі різноманітності і фундаментальні відмінності, є підставою мистецтва талановитих розробників для реалізації ефективних нейромереж.

Лабораторна робота № 4: Реалізація системи розпізнавання перцептронного типу для випадку двох класів.

Використовуючи основні можливості першої лабораторної роботи, створити систему розпізнавання перцептронного типу з нормованими векторами ознак числового типу. Система повинна забезпечити наступні дії:

- Ввести послідовно з графічних файлів образи тестової послідовності для двох класів та відобразити їх у відповідному графічному компоненті;
- Створення абсолютних та нормованих ознакових векторів для кожного зразка навчальної послідовності з відображенням абсолютних та нормованих ознакових значень;
- Задати початкові значення коефіцієнтів перцептрона Розенблата та поточний крок навчання;
- Обчислити вектор нев'язки та провести класифікацію за поточним станом перцептрона;
- В режимі навчання з вчителем підтвердити або спростувати результат класифікації;
- Провести корекцію перцептрона та відобразити його стан після корекції.

Оцінити збіжність процесу навчання.

Навчальна тема № 7: Нейронні мережі асоціативного та циклічного типу. ШНМ Хопфілда та Хеммінга.

Зміст теми:

1. Мережа Хопфілда.
2. Алгоритм функціонування мережі Хопфілда.
3. Модель двошарової нейронної мережі Хеммінга.

Теоретичні відомості.

1. Мережа Хопфілда.

Джон Хопфілд вперше представив свою асоціативну мережу у 1982 р. У Національній Академії Наук. На честь Хопфілда та нового підходу до моделювання, ця мережна парадигма згадується як мережа Хопфілда. Мережа базується на аналогії фізики динамічних систем. Початкові застосування для цього виду мережі включали асоціативну, або адресовану за змістом пам'ять та вирішували задачі оптимізації.

Мережа Хопфілда використовує три прошарки: вхідний, прошарок Хопфілда та вихідний прошарок. Кожен прошарок має однакову кількість нейронів. Входи прошарку Хопфілда під'єднані до виходів відповідних нейронів вхідного прошарку через змінні ваги з'єднань. Виходи прошарку Хопфілда під'єднуються до входів всіх нейронів прошарку Хопфілда, за винятком самого себе, а також до відповідних елементів у вихідному прошарку. В режимі функціонування, мережа скеровує дані з вхідного прошарку через фіксовані ваги з'єднань до прошарку Хопфілда. Прошарок Хопфілда коливається, поки не буде завершена певна кількість циклів, і біжучий стан прошарку передається на вихідний прошарок. Цей стан відповідає образу, вже запрограмованому у мережу.

Навчання мережі Хопфілда вимагає, щоб навчальний образ був представлений на вхідному та вихідному прошарках одночасно. Рекурсивний характер прошарку Хопфілда забезпечує засоби корекції всіх ваг з'єднань. Недвійкова реалізація мережі повинна мати пороговий механізм у передатній функції. Для правильного навчання мережі відповідні пари "вхід-вихід" мають відрізнятися між собою.

Якщо мережа Хопфілда використовується як пам'ять, що адресується за змістом вона має два головних обмеження. По-перше, число образів, що можуть бути збережені та точно відтворені є строго обмеженим. Якщо зберігається занадто багато параметрів, мережа може збігатись до нового неіснуючого образу, відмінному від всіх запрограмованих образів, або не збігатись взагалі. Межа ємності пам'яті для мережі приблизно 15% від числа нейронів у прошарку Хопфілда. Другим обмеженням парадигми є те, що прошарок Хопфілда може стати нестабільним, якщо навчальні приклади є

занадто подібними. Зразок образу вважається нестабільним, якщо він застосовується за нульовий час і мережа збігається до деякого іншого образу з навчальної множини. Ця проблема може бути вирішена вибором навчальних прикладів більш ортогональних між собою.

Структурна схема мережі Хопфілда приведена на рис. 1.

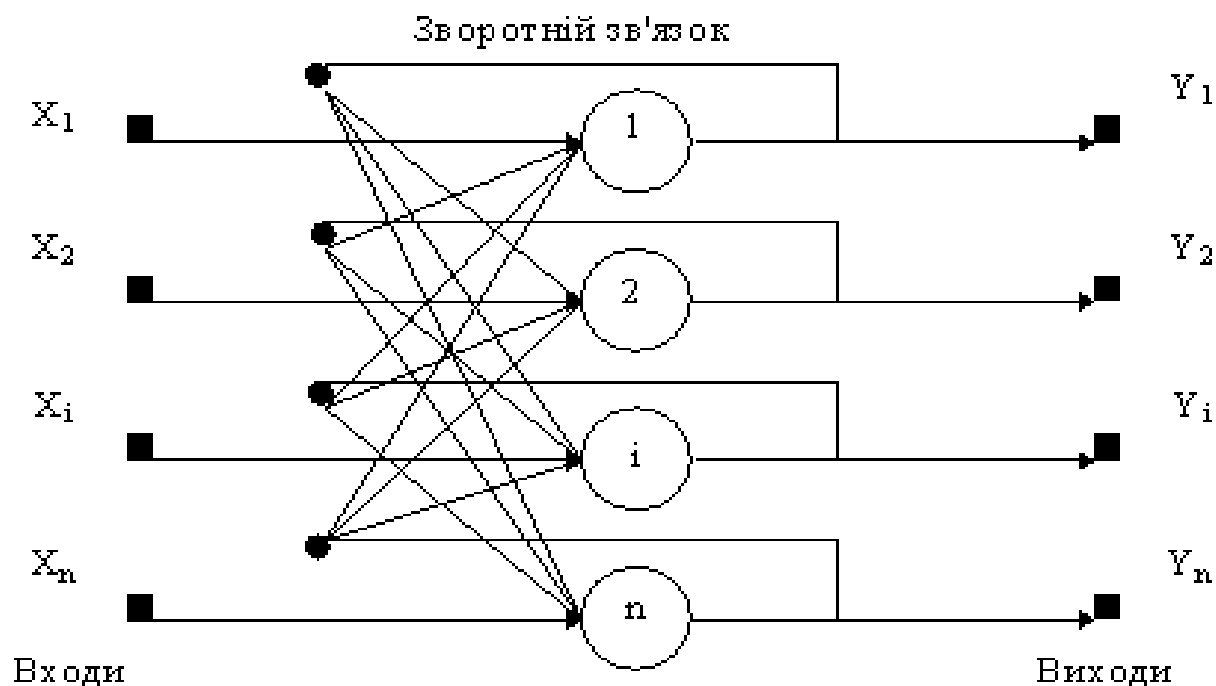


Рис. 1. Структурна схема мережі Хопфілда.

Задача, розв'язувана даною мережею в якості асоціативної пам'яті, як правило, формулюється таким чином. Відомий деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують якийсь об'єкти або характеристики процесів), вважають зразковим. Мережа повинна вміти з зашумленого сигналу, поданого на її вхід, виділити ("пригадати" по частковій інформації) відповідний зразок або "дати висновок" про те, що вхідні дані не відповідають жодному із зразків. У загальному випадку, будь-який сигнал може бути описаний вектором x_1, x_i, x_n, \dots , n - число нейронів у мережі і величина вхідних і вихідних векторів. Кожний елемент x_i дорівнює або $+1$, або -1 . Позначимо вектор, що описує k -ий зразок, через X_k , а його компоненти, відповідно, - x_{ik} , $k=0, \dots, m-1$, m - число зразків. Якщо мережа розпізнає (або "пригадає") якийсь зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто $Y = X_k$, де Y - вектор вихідних значень мережі: y_1, y_i, y_n . У протилежному випадку, вихідний вектор не співпаде з жодним зразковим. Якщо, наприклад, сигнали являють собою якесь зображення, то, відобразивши у графічному виді дані з виходу мережі, можна буде побачити картинку, що цілком збігається з однією зі зразкових (у випадку успіху) або ж "вільну імпровізацію" мережі (у випадку невдачі).

2. Алгоритм функціонування мережі Хопфілда.

Алгоритм функціонування мережі

1. На стадії ініціалізації мережі вагові коефіцієнти синапсів встановлюються таким чином:

$$w_{ij} = \begin{cases} \frac{\sum_{k=1}^m x_i^k x_j^k}{m}, & i \neq j \\ 0, & i = j \end{cases}$$

Тут i і j - індекси, відповідно, предсинаптичного і постсинаптичного нейронів; x_{ik} , x_{jk} - i -ий і j -ий елементи вектора k -ого зразка.

2. На входи мережі подається невідомий сигнал (t - номер ітерації). Його поширення безпосередньо встановлює значення виходів:

$$Y_i(0) = x_i, \quad i = 0 \dots n-1,$$

Тому позначення на схемі мережі вхідних сигналів у явному виді носить чисто умовний характер. Нуль у скобці справа від y_i означає нульову ітерацію в циклі роботи мережі.

3. Розраховується новий стан нейронів :

$$S_j(t+1) = \sum_{i=1}^n w_{ij} y_i(t), \quad j=0 \dots n-1.$$

І нові значення виходів:

$$y_j(t+1) = f[S_j(t+1)]$$

Де f - передатна функція у виді порогової, приведена на рис. 10.

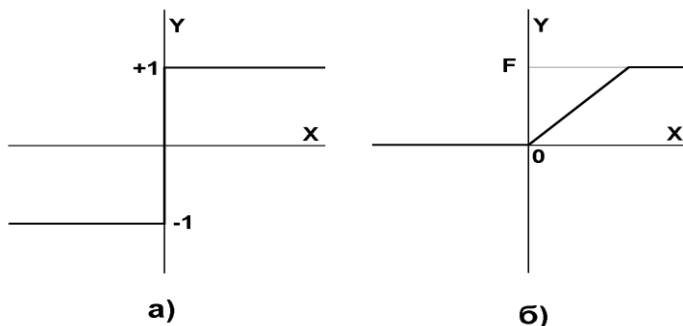


Рис. 10. Передаточні функції.

4. Перевіряємо чи змінилися вихідні значення виходів за останню ітерацію. Якщо так - перехід до пункту 2, інакше (якщо виходи стабілізувались) - кінець. При цьому вихідний вектор являє собою зразок, що найкраще відповідає вхідним даним.

Іноді мережа не може провести розпізнавання і видає на виході неіснуючий образ. Це пов'язано з проблемою обмеженості можливостей мережі. Для мережі Хопфілда число запам'ятованих образів m не повинно перевищувати величини, приблизно рівної $0.15 \cdot n$. Крім того, якщо два образи А і Б сильно схожі, вони, можливо, будуть викликати в мережі перехресні асоціації, тобто пред'явлення на входи мережі вектора А призведе до появи на її виходах вектори Б і навпаки.

4. Модель двошарової нейронної мережі Хеммінга.

Мережа Хеммінга (*Hamming*) є розширенням мережі Хопфілда. Ця мережа була розроблена Річардом Ліппманом (*Richard Lippman*) у середині 80-х рр. Мережа Хеммінга реалізує класифікатор, що базується на найменшій похибці для векторів двійкових входів, де похибка визначається відстанню Хеммінга. Відстань Хеммінга визначається як число бітів, які відрізняються між двома відповідними вхідними векторами фіксованої довжини. Один вхідний вектор є незашумленим прикладом образу, інший є спотвореним образом. Вектор виходів навчальної множини є вектором класів, до яких належать образи. У режимі навчання вхідні вектори розподіляються до категорій для яких відстань між зразковими вхідними векторами та біжучим вхідним вектором є мінімальною.

Мережа Хеммінга має три прошарки: вхідний прошарок з кількістю вузлів, скільки є окремих двійкових ознак; прошарок категорій (прошарок Хопфілда), з кількістю вузлів, скільки є категорій або класів; вихідний прошарок, який відповідає числу вузлів у прошарку категорій.

Мережа є простою архітектурою прямого поширення з вхідним рівнем, повністю під'єднаним до прошарку категорій. Кожен елемент обробки у прошарку категорій є зворотно під'єднаним до кожного нейрона у тому ж самому прошарку і прямо під'єднаним до вихідного нейрону. Вихід з прошарку категорій до вихідного прошарку формується через конкуренцію.

Навчання мережі Хеммінга є подібним до методології Хопфілда. На вхідний прошарок надходить бажаний навчальний образ, а на виході вихідного прошарку надходить значення бажаного класу, до якого належить вектор. Вихід містить лише значення класу до якої належить вхідний вектор. Рекурсивний характер прошарку Хопфілда забезпечує засоби корекції всіх ваг з'єднань.

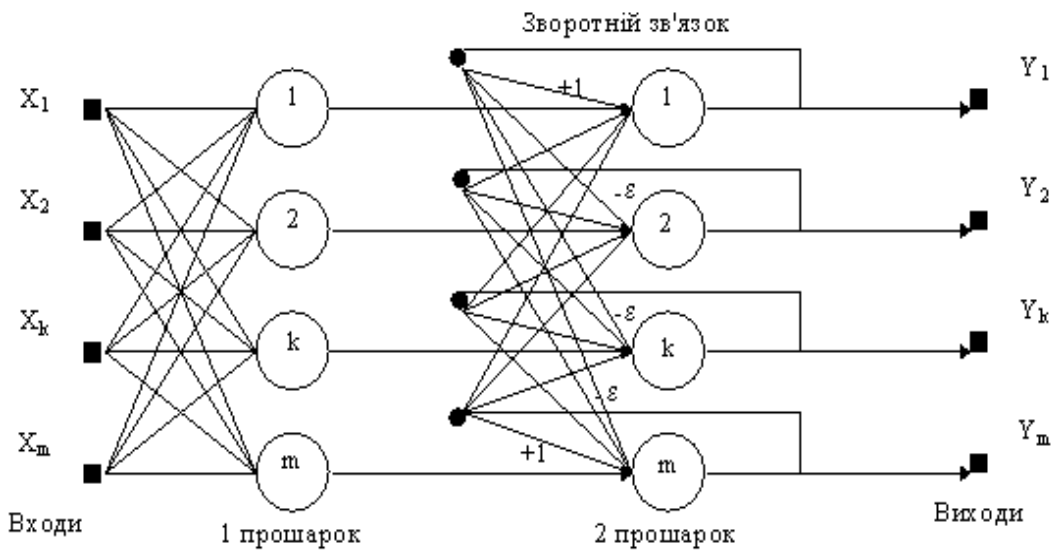


Рис. 11. Структурна схема мережі Хемінга.

Алгоритм функціонування мережі Хеммінга:

1. На стадії ініціалізації ваговим коефіцієнтам першого прошарку і порогу передатної функції присвоюються такі значення:

$$W_{ik} = x_i^k / 2, \quad i=0 \dots n-1, \quad k=0 \dots m-1$$

$$B_k = n / 2, \quad k = 0 \dots m-1 .$$

Тут x_i^k - i -ий елемент k -ого зразка.

Вагові коефіцієнти гальмуючих синапсів у другому прошарку беруть рівними деякій величині $0 < \nu < 1/m$. Синапс нейрона, пов'язаний із його ж виходом має вагу $+1$.

2. На входи мережі подається невідомий вектор $x_1, x_i, x_n \dots$. Розраховуються стани нейронів першого прошарку (верхній індекс у скобках указує номер прошарку):

$$y_j^{(1)} = S_j^{(1)} = \sum_{i=1}^n w_{ij} x_i + b_j, \quad j=0 \dots m-1.$$

Після цього отримання значення ініціалізують значення виходів другого прошарку:

$$Y_j^{(2)} = y_j^{(1)}, \quad j = 0 \dots m-1.$$

3. Обчислюються нові стани нейронів другого прошарку:

$$S_j^{(2)}(t+1) = y_j(t) - \varepsilon \sum_{k=1}^m y_k^{(2)}(t), k \neq j, j = 1 \dots m$$

І значення їх виходів:

$$y_j^{(1)}(t+1) = f|S_j^{(2)}(t+1)|.$$

Передатна функція f має вид порога, причому величина b повинна бути достатньо великою, щоб будь-які можливі значення аргументу не призводили до насичення.

4. Перевіряється, чи змінилися виходи нейронів другого прошарку за останню ітерацію. Якщо так - перейти до кроку 3. Інакше - кінець.

Роль першого прошарку є умовною: скориставшись один раз на першому кроці 1 значеннями його вагових коефіцієнтів, мережа більше не вертається до нього, тому перший прошарок може бути взагалі виключений із мережі. Мережа Хеммінга має ряд переваг над мережею Хопфілда. Вона реалізує оптимальний класифікатор мінімуму похибки, якщо похибки вхідних бітів є випадковими та незалежними. Для функціонування мережі Хеммінга потрібна менша кількість нейронів, оскільки середній прошарок вимагає лише один нейрон на клас, замість нейрону на кожен вхідний вузол. І, нарешті, мережа Хеммінга не страждає від неправильних класифікацій, які можуть трапитись у мережі Хопфілда. В цілому, мережа Хеммінга є як швидшою, так і точнішою за мережу Хопфілда.

Лабораторна робота № 5: Реалізація системи розпізнавання на базі нейронної мережі асоціативного та рекурентно-асимптотичного типу.

Розробити систему розпізнавання на базі штучної нейронної мережі Хопфілда з наступними діями (за варіантом лабораторної роботи №1):

- введення зразкових зображень образів трьох класів та відображення у відповідних графічних компонентах;
- обчислення абсолютних ознакових векторів та відображення ознакових значень;
- для штучної нейронної мережі Хопфілда провести бінаризацію еталонних ознакових векторів зі значеннями -1 та 1; для мережі Хеммінга провести нормування ознакових еталонних векторів;

- обчислити матрицю коефіцієнтів ШНМ Хопфілда або вагові коефіцієнти першого шару зв'язків у мережі Хеммінга. Відобразити відповідні матриці;
- ввести зображення невідомого вектора, обчислити його ознакові вектори відповідно до типу мережі;
- провести класифікацію за методом Хопфілда або Хеммінга;
- проаналізувати процес класифікації.

Навчальна тема № 8: Однопрохідні багатощарові ШНМ. Алгоритми навчання. Алгоритм оберненого розповсюдження помилки.

Зміст теми:

1. Нейромережа зворотного поширення похибки (Back Propagation)
2. Реалізація алгоритмів навчання методом оберненого розповсюдження помилки. Оптимізаційний підхід до навчання ШНМ.
3. Оцінка функціональних можливостей.

Теоретичні відомості.

1. Нейромережа зворотного поширення похибки (Back Propagation)

Архітектура *feedforward backpropagation* була розроблена на початку 1970-х років декількома незалежними авторами: Вербор (*Werbos*); Паркер (*Parker*); Румельгарт (*Rumelhart*), Хінтон (*Hinton*) та Вільямс (*Williams*). На даний час, парадигма *backpropagation* найбільш популярна, ефективна та легка модель навчання для складних, багатощарових мереж. Вона використовується у різних типах застосувань і породила великий клас нейромереж з різними структурами та методами навчання.

Типова мережа *backpropagation* має вхідний прошарок, вихідний прошарок та принаймні один прихований прошарок. Теоретично, обмежень відносно числа прихованих прошарків не існує, але практично застосовують один або два (рис. 1).

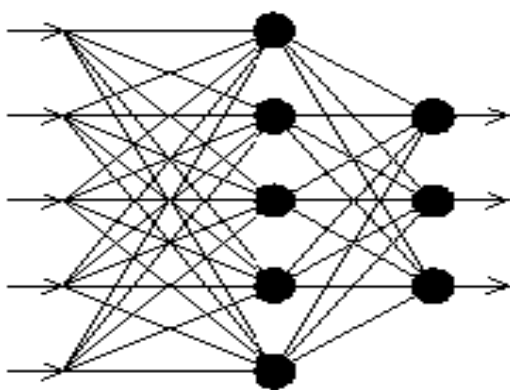


Рис. 3. Багатощаровий перцептрон.

Нейрони організовані в шарову структуру з прямою передачею сигналу. Кожний нейрон мережі продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення. Мережа може моделювати функцію практично будь якої складності, причому число прошарків і число нейронів у кожному прошарку визначають складність

функції. Визначення числа проміжних прошарків і числа нейронів в них є важливим при моделюванні мережі. Більшість дослідників та інженерів, застосовуючи архітектуру до визначених проблем використовують загальні правила, зокрема:

1. Кількість входів та виходів мережі визначаються кількістю вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо. На відміну від зовнішніх прошарків, число нейронів прихованого прошарку $n_{\text{прих}}$ обирається емпіричним шляхом. В більшості випадків достатня кількість нейронів становить $n_{\text{прих}} = n_{\text{вх}} + n_{\text{вих}}$, де $n_{\text{вх}}$, $n_{\text{вих}}$ - кількість нейронів у вхідному і, відповідно, у вихідному прошарках.
2. Якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого прошарку повинна також збільшитись.
3. Якщо процес, що моделюється, може розділятися на багато етапів, потрібен додатковий прихований прошарок (прошарки). Якщо процес не розділяється на етапи, тоді додаткові прошарки можуть допустити перезапам'ятовування і, відповідно, невірне загальне рішення.

Після того, як визначено число прошарків і число нейронів в кожному з них, потрібно знайти значення для синаптичних ваг і порогів мережі, які спроможні мінімізувати похибку спродукованого результату. Саме для цього існують алгоритми навчання, де відбувається підгонка моделі мережі до наявних навчальних даних. Похибка для конкретної моделі мережі визначається шляхом проходження через мережу всіх навчальних прикладів і порівняння спродукованих вихідних значень з бажаними значеннями. Множина похибок створює функцію похибок, значення якої можна розглядати, як похибку мережі. В якості функції похибок найчастіше використовують суму квадратів похибок. Для кращого розуміння алгоритму навчання мережі *Back Propagation* потрібно роз'яснити поняття поверхні станів. Кожному значенню синаптичних ваг і порогів мережі (вільних параметрів моделі кількістю N) відповідає один вимір в багатовимірному просторі. $N+1$ -ий вимір відповідає похибці мережі. Для різноманітних сполучень ваг відповідну похибку мережі можна зобразити точкою в $N+1$ -вимірному просторі, всі ці точки утворюють деяку поверхню - поверхню станів. Мета навчання нейромережі полягає в знаходженні на багатовимірній поверхні найнижчої точки.

Поверхня станів має складну будову і досить неприємні властивості, зокрема, наявність локальних мінімумів (точки, найнижчі в своєму певному околі, але вищі від глобального мінімуму), пласкі ділянки, сідлові точки і довгі вузькі яри. Аналітичними засобами неможливо визначити розташування глобального мінімуму на поверхні станів, тому навчання нейромережі по суті полягає в дослідженні цієї поверхні. Відштовхуючись від початкової конфігурації ваг і порогів (від випадково обраної точки на поверхні), алгоритм навчання поступово відшукує глобальний мінімум. Обчислюється вектор градієнту поверхні похибок, який вказує напрямок найкоротшого спуску по поверхні з заданої точки. Якщо трошки просунутись по ньому, похибка

зменшиться. Зрештою алгоритм зупиняється в нижній точці, що може виявитись лише локальним мінімумом (в ідеальному випадку - глобальним мінімумом). Складність тут полягає у виборі довжини кроків. При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути рішення, або піти в неправильному напрямку. При маленькому кроці, правильний напрямок буде виявлений, але зростає кількість ітерацій. На практиці розмір кроку береться пропорційним крутизни схилу з деякою константою - швидкістю навчання. Правильний вибір швидкості навчання залежить від конкретної задачі і здійснюється дослідним шляхом. Ця константа може також залежати від часу, зменшуючись по мірі просування алгоритму. Алгоритм діє ітеративно, його кроки називаються епохами. На кожній епосі на вхід мережі по черзі подаються всі навчальні приклади, вихідні значення мережі порівнюються з бажаними значеннями і обчислюється похибка. Значення похибки, а також градієнту поверхні станів використовують для корекції ваг, і дії повторюються. Процес навчання припиняється або коли пройдена визначена кількість епох, або коли похибка досягає визначеного рівня малості, або коли похибка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій останову).

2. Реалізація алгоритмів навчання методом оберненого розповсюдження помилки. Оптимізаційний підхід до навчання ШНМ.

За методом найменших квадратів, функція мінімізації загальної помилки мережі подається формулою:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2, \quad (1)$$

де $y_{j,p}^{(N)}$ - реальні вихідні значення j -го нейрона останнього N -го шару нейронної мережі при аналізі p -го образу навчальної послідовності;

$d_{j,p}$ - вірне (бажане) значення вказаного нейрона.

Сумування ведеться по всім нейронам вихідного шару для всіх зразків навчальної послідовності. Зміна вагових значень нейронів проводиться по формулі:

$$\Delta w_{i,j}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{i,j}}, \quad (2)$$

де $w_{i,j}$ - вагове значення синапсичного зв'язку, що з'єднує i -ий нейрон шару $n-1$ з j -им нейроном шару n ;

η - крок навчання, вибирається на інтервалі $0 < \eta < 1$.

Частинні похідні обчислюються за формулою:

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{i,j}} \quad (3)$$

Тут y_j - вихід j -го нейрона, а s_j - виважена сума його вхідних сигналів.

Для сигмоїдальної функції активації нейрона її похідна здійснюється за формулою:

$$\frac{dy}{ds} = \frac{-b \cdot e^{-bs}}{(1 + e^{-bs})^2} \quad (4)$$

а для обчислення першого множника у формулі (3) використовується вираз

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot w_{j,k}^{(n+1)} \quad (5)$$

В виразі (5) сумування проводиться по ваговим коефіцієнтам наступного, $n+1$ -го шару.

Введенням нової змінної

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \quad (6)$$

Отримується рекурентна формула для обчислення величин $\delta_j^{(n)}$ шару n з відповідних величин $\delta_j^{(n+1)}$ старшого, наступного шару $n+1$:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{j,k}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j} \quad (7)$$

Формула (3.7) застосовується для всіх внутрішніх шарів, а для вихідного шару застосовується формула

$$\delta_j^{(n)} = (y_j^{(N)} - d_j) \cdot \frac{dy_j}{ds_j} \quad (8)$$

В результаті отримується рекурентне відношення для обчислення змін вагових синапсичних коефіцієнтів:

$$\Delta w_{i,j}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \quad (9)$$

Або (для досягнення інерційності):

$$\Delta w_{i,j}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{i,j}^{(n)}(t-1) + (1-\mu) \delta_j^{(n)} \cdot y_i^{(n-1)}) \quad (10)$$

де μ - коефіцієнт інертності, $0 < \mu < 1$;

t - крок навчання.

За такою математичною моделлю, алгоритм зворотного розповсюдження похибки має вигляд:

1. Всім ваговим коефіцієнтам надаються малі початкові значення на інтервалі (0,1).

2. Створюється цикл для вибору чергових зразків з навчальної послідовності.
3. На вхід нейронної мережі подається чергових образ навчальної послідовності. Для всіх нейронів мережі обчислюються внутрішні стани за формулою

$$s_j^{(n)} = \sum_{i=0}^M y_i^{(n-1)} \cdot w_{i,j}^{(n)}, \quad (11)$$

де M - розмір вхідного вектору. Нульова компонента вхідного вектора одинична и визначає загальне зміщення суми.

$y_i^{(n-1)}$ - вихідні значення нейронів попереднього шару; для першого шару замінюються компонентами вхідного вектора.

4. Обчислюються вихідні значення всіх нейронів за формулою

$$y_j^{(n)} = f(s_j^{(n)}), \quad (12)$$

де $f(s)$ сигмоїдальна функція.

5. Для останнього шару нейронів вказиваються правильні (бажані) значення вихідних сигналів y_j , за формулою (8) обчислюються величини $\delta_j^{(N)}$.
6. За формулою (9) або (10) обчислюються зміни вагових коефіцієнтів вихідного шару $\Delta w_{i,j}^{(N)}$.
7. За формулою (7) и (9) або (7) и (10) обчислюються величини $\delta_j^{(n)}$ и зміни вагових коефіцієнтів всіх внутрішніх шарів послідовно від передостаннього до першого шару.
8. Вагові коефіцієнти всіх шарів перераховуються за формулою

$$\Delta w_{i,j}^{(n)}(t) = w_{i,j}^{(n)}(t-1) + \Delta w_{i,j}^{(n)}(t), \quad (13)$$

де t - крок навчання.

Процес навчання продовжується, поки функція сумарної похибки нейронної мережі не задовольнить наперед задані параметри точності та надійності.

Навчальна тема № 9: Еволюційне моделювання та генетичний алгоритм в задачах штучного інтелекту.

Зміст теми:

1. Поняття еволюційного програмування.
2. Структура генетичного алгоритму.
3. Генетичні операції: кросовер, мутація, відбір.
4. Особливості реалізації. Оцінка збіжності. Модифікації алгоритму.

Теоретичні відомості.

1. Поняття еволюційного програмування.

Генетичний алгоритм (*genetic algorithm*) — це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Особливістю генетичного алгоритму є акцент на використання оператора "схрещення", який виконує операцію рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі.

2. Структура генетичного алгоритму.

Задача кодується таким чином, щоб її вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи допущені до схрещення (*селекція*). До осіб застосовується "генетичні оператори" (в більшості випадків це оператор схрещення (*crossover*) і оператор мутації (*mutation*), створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (*покоління*), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

- знаходження глобального, або надоптимального рішення;
- вичерпання числа поколінь, що відпущені на еволюцію;
- вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми можуть використатися для пошуку рішень в дуже великих і тяжких просторах пошуку.

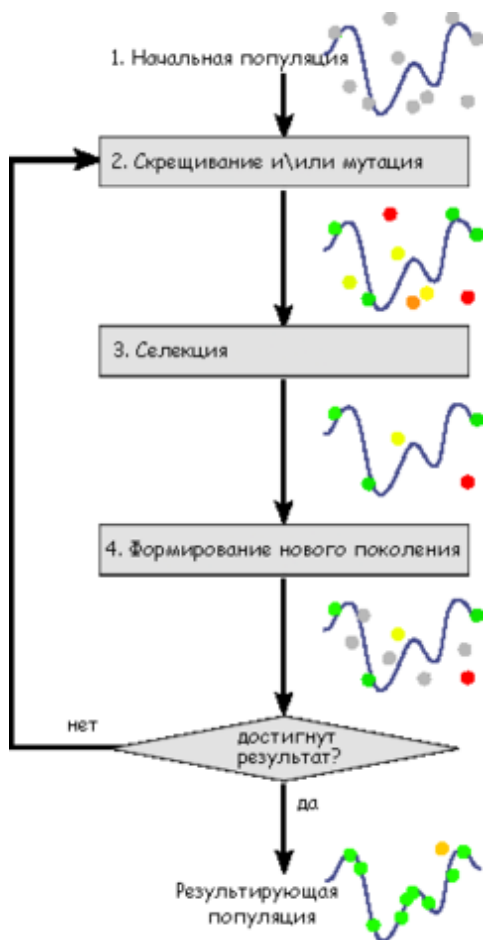


Схема роботи генетичного алгоритму

Етапи генетичного алгоритму

Можна виділити такі етапи генетичного алгоритму:

1. Створення початкової популяції:
2. Обчислення функції пристосованості для осіб популяції (оцінювання)
3. Повторювання до виконання критерію зупинки алгоритму:
 1. Вибір індивідів із поточної популяції (селекція)
 2. Схрещення або/та мутація
 3. Обчислення функції пристосовуваності для всіх осіб
 4. Формування нового покоління

Створення початкової популяції

Перед першим кроком необхідно отримати з життя або випадковим чином створити деяку початкову популяцію. Навіть якщо популяція здається зовсім не конкурентною, генетичний алгоритм все одно достатньо швидко переведе її в

придатну для життя популяцію. Таким чином, на першому кроці можна не особливо старатися зробити надто вже пристосованих осіб, достатньо, щоб вони відповідали формату осіб популяції, і на них можна було вирахувати функції пристосованості (Fitness). Наслідком першого кроку є популяція N , що налічує N осіб.

Відбір

На етапі відбору необхідно із всієї популяції вибрати її певну долю, яка залишиться в «живих» на цьому етапі популяції. Є декілька способів провести відбір. Ймовірність виживання особи h повинна залежати від значення її пристосованості $Fitness(h)$. Сама ж доля відібраних s зазвичай є параметром генетичного алгоритму, і її просто задають заздалегідь. Внаслідок відбору із N осіб популяції N повинні залишитись sN осіб, які ввійдуть в наступну популяцію N' . Решта осіб «загине».

Розмноження

Розмноження в генетичних алгоритмах зазвичай статеве - щоб «народити» нащадка, необхідно декілька батьків, зазвичай потрібна участь двох. Розмноження в різних алгоритмах описується по різному - воно, звісно, залежить від формату осіб. Головна вимога до розмноження - щоб нащадок чи нащадки мали можливість успадкувати риси всіх батьків, «змішавши» їх якимось достатньо розумним чином.

Розмноження або оператор рекомбінації застосовують відразу ж після оператора відбору батьків для отримання нових особин-нащадків. Сенса рекомбінації полягає в тому, що створені нащадки повинні наслідувати генну інформацію від обох батьків. Особи для розмноження зазвичай вибираються із всієї популяції N , а не із тих, що вижили на першому кроці (хоча останній варіант теж має право на існування). Справа в тому, що головна проблема генетичних алгоритмів - нестача різноманітності (diversity) в особах. Достатньо швидко виділяється єдиний генотип, який представляє собою локальний максимум і згодом всі елементи популяції програють йому в відборі, і вся популяція "забивається" копіями цієї особи. Існують різні способи боротьби із таким небажаним ефектом; один з них - вибір для розмноження не з самих "пристосованих", а взагалі зі всіх осіб.

Мутації

До мутацій відноситься все те ж, що і до розмноження: є деяка доля мутантів m , що є параметром генетичного алгоритму, і на кроці мутацій необхідно вибрати mN осіб, а згодом змінити їх згідно з заздалегідь заданими операціями мутації.

Лабораторна робота № 6: Реалізація алгоритмів оптимізації нетрадиційної задачі методом генетичного алгоритму.

Створити математичну модель та реалізувати демонстраційну програму оптимізації функції двох змінних зі складною областю визначення методом генетичного алгоритму з наступними функціями:

- Введення параметрів генетичного алгоритму (розмір початкової популяції, кількість мутацій, кількість пар для кросовера);
- Ввести систему обмежень для створення не опуклої області визначення. Відобразити область визначення в тривимірній системі координат;
- Створення початкової популяції та відображення її екземплярів в тривимірній системі координат;
- Для поточного кроку регенерації показати:
 - Значення цільової функції кожного екземпляру популяції;
 - Екземпляри поточної вибірки;
 - Нові отримані екземпляри;
- Провести операції кросовера та мутації, показати нові екземпляри популяції;
- Провести селекцію, вилучити неконкурентних екземплярів;
- На кожному кроці регенерації показати у впорядкованому порядку числові значення екземплярів популяції та їх оцінку та відобразити на зображенні та у числовому вигляді розв'язок задачі оптимізації.

Примітка:

Для кращої демонстрації процесів, що відбуваються у генетичному алгоритмі, рекомендовано вибрати функцію, яка у області визначення має декілька точок локального мінімуму.

Рекомендована література

1. Нікольський Ю.В., Пасічник В.В., Щербина Ю.М. Системи штучного інтелекту : навчальний посібник. Київ : Магнолія, 2021. 280 с.
2. Гончаров О.А., Васильєва Л.В., Юнда А.М. Чисельні методи розв'язання прикладних задач : навч. посібн. Суми : Сумський державний університет, 2020, 142 с.
3. Шаховська Н.Б., Камінський Р.М., Вовк О.Б. Системи штучного інтелекту : навчальний посібник. Львів : Львівська політехніка. 2018. 392 с.
4. Троцько В.В. Методи штучного інтелекту: навчально-методичний і практичний посібник / В.В. Троцько. К. : Університет "КРОК", 2020. 86 с.
5. Савченко А.С., Синельников О.О. Методи та системи штучного інтелекту: навчальний посібник. Київ : Національний авіаційний інститут. 2017. 188 с.
6. Зайченко Ю.П. Основи проектування інтелектуальних систем. Навчальний посібник. — Київ, Видавничий Дім "Слово", 2004 р. - 352 с.
7. Wasserman, Philip D, Neural Computing: Theory and Practice, First Edition, Hygiene, Colorado, 1989, 230 p.
8. Фратавчан В.Г., Фратавчан Т.М. Нейронні мережі: методичні вказівки та завдання для лабораторних робіт, Чернівці: ЧНУ, 2022. 32 с.

Інформаційні ресурси

1. Учбовий курс "Системи штучного інтелекту"
<https://web.archive.org/web/20160907035551/http://victoria.lviv.ua/html/ai/>
2. Earl Hunt, Artificial Intelligence
https://books.google.com.ua/books?hl=uk&lr=&id=9y2jBQAAQBAJ&oi=fnd&pg=PP1&dq=Hunt+Artificial+intelligence&ots=uPXxXi9XqB&sig=0kXUTxTZGJHsiWRRlyBPnG6R59E&redir_esc=y#v=onepage&q=Hunt%20Artificial%20intelligence&f=false
3. Книга Ф.Уосермена «Нейрокомп'ютерна техніка: Теорія і практика»
<https://web.archive.org/web/20090531081836/http://www.victoria.lviv.ua/html/wosserman/index.htm>