

Лабораторна робота №3.

Використання еволюційних алгоритмів

Мета роботи. Ознайомитись із принципами еволюційних алгоритмів як одного з напрямів штучного інтелекту. Отримати практичні навички використання еволюційних алгоритмів для вирішення оптимізаційних задач.

Завдання

1. Ознайомитися з основними теоретичними відомостями та рекомендованою літературою за темою роботи.

2. Вибрати та виконати **ОДИН** з двох поданих варіантів нижче.

3. Оформити звіт. Структура звіту: титульний аркуш, мета роботи, покроковий опис ходу роботи, приклад програмної реалізації, скріншоти з демонстрацією результатів виконання програми, висновки.

Варіант 1.

Виконати програмну реалізацію будь-якого еволюційного алгоритму (вибрати довільно генетичний алгоритм, еволюційну стратегію, алгоритм еволюційного програмування, алгоритм генетичного програмування тощо) для визначення набору гіперпараметрів обраної вами моделі машинного навчання. Можна використати готову бібліотеку, де вже реалізовано обраний еволюційний алгоритм, але знати як він працює.

Модель машинного навчання та предметну область для її застосування обрати довільно (наприклад, класифікація текстів, прогнозування цін на нерухомість, тощо).

Фітнес-функцією може бути точність, середньоквадратична помилка, або будь-яка інша метрика, яка відображає продуктивність моделі на валідаційному наборі даних. Використати еволюційний алгоритм для оптимізації гіперпараметрів.

Алгоритм повинен відбирати підмножину гіперпараметрів, генерувати розв'язки за допомогою мутації, оцінювати їхню пристосованість та оновлювати найкращі розв'язки для наступної ітерації. Сформулювати висновки щодо результатів використання еволюційного алгоритму для вирішення обраної вами задачі.

Варіант 2.

Виконати програмну реалізацію еволюційних операторів відповідно до свого варіанту в таблиці 1. Варіант обирати згідно зі своїм порядковим номером у загальному списку групи.

Таблиця 1 – Еволюційні оператори для виконання завдання

№ варіанту			Еволюційні оператори		
			<i>Відбір</i>	<i>Схрещування</i>	<i>Мутація</i>
1	15		SUS	Однорідне	Bit Flip
2	16		Ранжування	Багатоточкове	Випадкове скидання
3	17		Рулетка	Однорідне	Заміна
4	18		Випадковий	Схрещування Девіса (OX1)	Scramble
5	19		Турнірний	Ціла арифметична рекомбінація	Інверсійна
6	20		SUS	Однорідне	Bit Flip
7	21		Ранжування	Багатоточкове	Випадкове скидання
8	22		Рулетка	Однорідне	Заміна
9	23		Випадковий	Схрещування Девіса (OX1)	Scramble
10	24		Турнірний	Ціла арифметична рекомбінація	Інверсійна
11	25		SUS	Однорідне	Випадкове скидання
12	26		Ранжування	Багатоточкове	Bit Flip
13	27		Рулетка	Однорідне	Заміна
14	28		Випадковий	Схрещування Девіса (OX1)	Інверсійна

Інші параметри, необхідні для реалізації, обрати самостійно. Вибір параметрів аргументувати.

Провести експерименти з реалізованими операторами на **двох** задачах з:

- Задача мінімізації функції Розенброка на діапазоні $[-5, 5]$ (для перевірки ефективності різних видів схрещення та мутації).

- Задача рюкзака (для перевірки ефективності різних видів відбору та схрещення).

- Задача класифікації даних (для перевірки ефективності різних видів відбору та мутації).

Провести аналіз результатів експериментів та зробити висновки щодо ефективності використання обраних еволюційних операторів для розв'язання обраних задач.

Задача мінімізації функції Розенброка - [Опис задачі](#)

Задача рюкзака – [Опис задачі](#)

Задача класифікації даних - [Опис задачі](#)

Лабораторна робота №4.

Розробка Retrieval-Augmented Generation (RAG) системи

Мета роботи. Ознайомитися з архітектурою RAG. Інтегрувати LLM із зовнішнім джерелом даних. Оцінити якість відповідей агента та провести аналіз.

Завдання

1. Ознайомитися з основними теоретичними відомостями та рекомендованою літературою за темою роботи.

2. Вибрати корпус документів (наукові статті (arXiv), Wikipedia-добірка, внутрішні документи (наприклад, набір PDF/CSV) тощо).

3. Виконати попередню обробку даних (очистити текст від шумів; розбити на фрагменти (чанки) (наприклад, по 500–1000 слів із перекриттям) і т.д.)

4. Створити векторну базу. Обрати бібліотеку для вбудовувань.

5. Побудувати векторні представлення та зберегти їх у векторній базі.
6. Реалізувати пошук: на вхід запиту → топ-k найближчих chunks.
7. Обрати довільну LLM.
8. Реалізувати простий RAG-пайплайн - користувацький запит → пошук релевантних документів → об'єднання контексту → генерація відповіді.
9. Реалізувати можливість зберігання історії діалогу.
10. Додати ранжування результатів за релевантністю (наприклад, використати косинусну подібність або інші міри оцінки подібності).
11. Виконати оцінку якості роботи розробленої системи:
 - Скласти набір із 10–15 контрольних запитань;
 - Запустити систему з різними параметрами;
 - Оцінити якість відповідей за критеріями: точність (чи інформація правильна); повнота (чи враховано всі релевантні частини); узгодженість (чи логічна відповідь).
12. Оформити звіт. Структура звіту: титульний аркуш, мета роботи, покроковий опис ходу роботи, приклад програмної реалізації, скріншоти з демонстрацією результатів виконання програми, висновки.