МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

РОЗГОРТАННЯ ПРЕ-ТРЕНОВАНОЇ МОДЕЛІ КЛАССУ GPT В ЛОКАЛЬНОМУ СЕРЕДОВИЩІ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи № 2
з дисципліни «Штучний інтелект в ігрових застосунках»
для студентів бакалаврського рівня вищої освіти спеціальності 121
"Інженерія програмного забезпечення"

Львів -- 2025

Розгортання пре-тренованої моделі GPT в локальному середовищі: методичні вказівки до виконання лабораторної роботи №2 з дисципліни "Штучний інтелект в ігрових застосунках" для студентів першого (бакалаврського) рівня вищої освіти спеціальності 121 "Інженерія програмного забезпечення" . Укл.: О.Є. Бауск. -- Львів: Видавництво Національного університету "Львівська політехніка", 2025. -- 10 с.

Укладач: Бауск О.Є., к.т.н., асистент кафедри ПЗ

Відповідальний за випуск: Федасюк Д.В., доктор техн. наук, професор

Рецензенти: Федасюк Д.В., доктор техн. наук, професор

Задорожний І.М., асистент кафедри ПЗ

Тема роботи: Розгортання попередньо тренуваної моделі GPT в локальному середовищі.

Мета роботи: Ознайомитись з основами функціонування системи-обгортки для моделей глибокого навчання OLLAMA, навчитися розгортати навчені моделі.

Теоретичні відомості

Що таке Ollama?

Ollama — це інструмент з відкритим вихідним кодом, призначений для простого та швидкого розгортання моделей штучного інтелекту (AI), зокрема великих мовних моделей (LLM), у локальному середовищі. Він дозволяє користувачам легко завантажувати, зберігати, керувати та запускати попередньо треновані моделі без необхідності складних налаштувань чи глибоких знань у сфері машинного навчання.

Як працює Ollama?

Ollama працює як система-обгортка (wrapper), яка забезпечує зручний інтерфейс для взаємодії з моделями глибокого навчання. Він автоматизує процес завантаження моделей з віддалених репозиторіїв, їх зберігання, керування версіями та запуску. Ollama підтримує різні популярні моделі, такі як GPT, LLaMA, DeepSeek та інші, дозволяючи користувачам швидко розгорнути їх локально та використовувати для вирішення різноманітних задач, включаючи генерацію тексту, відповіді на запитання, створення чат-ботів тощо.

Що таке попередньо треновані моделі (pretrained models)?

Попередньо треновані моделі — це моделі машинного навчання, які вже були навчені на великих наборах даних для виконання певних загальних задач. Наприклад, мовні моделі, такі як GPT, тренуються на величезних об'ємах текстових даних з інтернету, книг, статей тощо. Після такого тренування модель здатна розуміти контекст, генерувати текст, відповідати на запитання та виконувати інші завдання, пов'язані з обробкою природної мови.

Користувачі можуть використовувати ці моделі без додаткового навчання або ж донавчати їх (fine-tuning) на власних специфічних наборах даних для покращення результатів у конкретних задачах.

Як зберігаються та керуються моделями в Ollama?

Ollama зберігає моделі у вигляді спеціальних контейнерів, які містять усі необхідні файли та параметри для роботи моделі. Користувачі можуть легко завантажувати моделі з віддалених репозиторіїв за допомогою простих команд (наприклад, `ollama pull <назва_моделі>`). Після завантаження модель зберігається локально, що дозволяє використовувати її без підключення до інтернету.

Ollama також надає зручні команди для перегляду списку доступних локально моделей (`ollama list`), видалення непотрібних моделей (`ollama rm <назва_моделі>`) та запуску моделей для виконання конкретних задач.

Таким чином, Ollama значно спрощує процес роботи з великими мовними моделями, роблячи їх доступними для широкого кола користувачів, включаючи тих, хто не має глибоких знань у сфері штучного інтелекту та машинного навчання.

УВАГА!

У випадку, коли ви бачите помилку "Warning: could not connect to a running Ollama instance" це означає, що локальний сервіс Ollama за якоюсь причиною не був запущений належним чином.

Щоб запустити локальну інстанцію Ollama, виконайте наступні кроки:

1. Відкрийте термінал або командний рядок.
2. Виконайте команду:

```
ollama serve
```

Це запустить локальну інстанцію Ollama, яка буде готова приймати запити. Не закривайте цей термінал, поки не завершите роботу з Ollama.

Висновок

Сучасні інструменти для розробки систем штучного інтелекту дозволяють розгортати навчені моделі в локальному середовищі. В даній роботі демонструється, як швидко і ефективно це зробити використовуючи тільки базові інструменти у відкритому доступі.

Хід роботи

1. Налаштування інструменту розгортання моделей машинного навчання Ollama.

1.1. Залежно від системи, на якій проводиться розгортання, встановити інструмент залежно від інструкцій на офіційному сайті: https://ollama.ai/.

На Windows:

```
https://ollama.com/download/windows
```

На Linux:

```
curl -fsSL https://ollama.com/install.sh | sh
```

1.2. Перевірити інсталяцію:

```
ollama --version
```

Має вивести встановлену версію системи розгортання моделей без помилок.

2. Встановлення модели LLM

Для задач даної лабораторної роботи ми хочемо використовувати локально модель натуральної генерації мови, яка виконує приблизно ті базові задачі, що, наприклад, широко відомий ChatGPT-o4-mini.

Зазвичай виконання подібної LLM моделі локально на власній машині практично неможливе, как як вона має міліарди параметрів і потребує вкрай потужного апаратного забезпечення.

Для вирішення цієї проблеми використаємо так звану дистільовану модель DeepSeek-R1 з 1 міліардом параметрів.

2.1. Що таке дистільована модель?

Дистільована модель — це модель, яка була тренувана на великому обсязі даних, але в результаті була зменшена до необхідного розміру. Це дозволяє зберігати модель у локальному середовищі і використовувати її для вирішення конкретних завдань.

Дистильовані моделі DeepSeek-R1 відносяться до так званих "щільних" моделей, особливістю яких є те, що з моделі було видалено незначну частину параметрів, які найменшевпливають на результат роботи моделі. Вони потребують менший обсяг пам'яті і можуть бути запущені на більш слабких машинах.

2.2. Скачати модель DeepSeek-R1:

УВАГА! Виконуйте даний етап тільки при наявності стабільного якісного інтернет з'єднання. Перевірте наявність кількох десятків ГБ вільного місця на диску.

Скачаємо архів з цією моделлю і розгорнемо його локально. В командній строці/терміналі:

```
ollama pull deepseek-r1
```

Перевіримо, що модель скачалась і зберігається локально:

```
ollama list
```

Ви маєте побачити інформацію про встановлену модель:

3. Використання моделі Deepseek.

Запустіть модель локально.

```
ollama run deepseek-r1
```

Ви маєте отримати командну строку, в якій можна задавати моделі промпти, спостерігати генерацію процесу формування вектора відповідей, і генерацію тексту моделлю.

3.1 Проведіть наступні експерименти з моделлю і внесіть результати в звіт:

3.1.1. Запустіть модель і надайте їй промпт:

```
Write a Python script to convert JPEG images to PNG.
```

Занотуйте скріншоти процесу генерації вектора відповідей. Оцініть час генерації.

Оцініть корисність результату.

3.1.2. Надайте промпт за власним вибором і занотуйте результати в звіт.

4. Технічні деталі про модель, що використовується.

4.1. Дослідіть модель DeepSeek-R1, що використовується в даній лабораторній роботі, або, якщо за якимись обставинами ви мали використати іншу модель, то опишіть її властивості.

Дослідіть документацію моделі: https://ollama.com/library/deepseek-r1

Скільки параметрів має модель, яку ви використовуєте?

Як, на вашу думку, зміняться результати роботи моделі, якщо використовувати модель з більшою кількістю параметрів? Меншою кількістю параметрів?

Які стадії/режими роботи моделі ви використовуєте саме на своїй локальній машині, при виконанні команд ollama run та надання текстових промптів?

Приклади режимів та стадій роботи практичної нейронної мережі:

- пре-тренування,
- тренування,
- валідація,
- дистіляція,
- розгортання,
- прогнозування,
- генерація результатів.

Внесіть відповіді на ці запитання в звіт.

4.2. Використання більш деградованої моделі.

Завантажте і запустіть модель DeepSeek-R1 з 1B параметрів замість 7B.

Якщо ви використовуєте іншу модель, то завантажте і запустіть модель з меншою кількістю параметрів.

Занотуйте якість генерації текстового вектора та швидкість виконання, внесіть висновки в звіт.

```
ollama pull deepseek-r1:1b
ollama run deepseek-r1:1b
```

Використайте промпти з попереднього завдання на моделі з меншою кількістю параметрів.

Порівняйте якість та швидкість відповідей. Зробіть висновки, як кількість параметрів впливає на швидкість виконання передбачення, на якість результату симуляції мисленного процесу, на появу галюцинацій у моделі.

УМОВА ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

1. Встановити систему розгортання моделей глибокого навчання Ollama.
2. Розгорнути локально LLM модель DeepSeek-R1 (Варіант з 1B параметрів).
3. Протестувати локальне розгортання моделі.
4. Дослідити налаштування моделей при локальному розгортанні, зрозуміти різницю між використанням онлайн- сервісів з LLM моделями та власного деплоймента.

ІНДІВІДУАЛЬНІ ВАРІАНТИ ЗАВДАННЯ

Створіти чат з локальною інсталяцією DeepSeek і використати наступні теми для розмови, залежно від номера в списку. -- див. пункт 3.1

ЗМІСТ ЗВІТУ

1. Тема та мета роботи
2. Теоретичні відомості
3. Постановка завдання
4. Хід виконання роботи:
    - Скріншоти процесу створення локальної інсталяції
    - Код та пояснення для створення моделі
    - Скріншоти інтерфейсу
5. Результати роботи
6. Висновки

КОНТРОЛЬНІ ПИТАННЯ

1. Що таке LLM моделі?
2. Що таке попередньо тренувані моделі (pretrained models)?
3. Що таке дистільовані моделі?
4. Що таке локальне розгортання моделі?
5. Який розмір моделі DeepSeek-R1 на вашому комп'ютері?
6. Які функції і задачі має інструмент розгортання моделей Ollama?
7. Яку роль відіграє кількість параметрів у роботі моделі, якості генерації тексту, та швидкості виконання?
8. Які переваги і недоліки локального розгортання моделі?

9. Які переваги і недоліки онлайн-сервісів з LLM моделями, таких як OpenAI/ChatGPT?
10. Які переваги і недоліки дистільованих моделей?

СПИСОК ЛІТЕРАТУРИ

1. Ollama
2. DeepSeek
3. LLM
4. Distilled models
5. DeepSeek-R1