УДК 004.8

СТВОРЕННЯ ТА НАВЧАННЯ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ З ВИКОРИСТАННЯМ БІБЛІОТЕК PYTHON

Р. Лук'яненко, група КБ-22-1
С. Смірнов, доцент
Н. Козірова, викладач
Р. Ткачук, асистент
Центральноукраїнський національний технічний університет

Зважаючи на актуальність та широке поширення штучного інтелекту(ШІ) у сучасному цифровому середовищі виникає необхідність у глибокому вивченні процесів створення та навчання мовних моделей. Незважаючи на доступність попередньо навчених моделей, розробка власних архітектур або адаптація існуючих під специфічні задачі вимагає знань про принципи токенізації, організацію навчального циклу, ефективне використання апаратних ресурсів (CPU, GPU, TPU), а також застосування спеціалізованих бібліотек, таких як Hugging Face Transformers, PyTorch і TensorFlow. У зв'язку з цим постає завдання дослідити сучасні інструменти створення та тренування моделей ШІ, визначити ключові етапи цього процесу, а також охарактеризувати пов'язані з ним виклики та можливості.

Навчання моделей штучного інтелекту (ШІ), зокрема трансформерних архітектур, вимагає глибокого розуміння як самої структури моделей, так і сучасних інструментів програмування. У роботі розглядаються ключові бібліотеки Python, які використовуються для розробки та навчання моделей ШІ, а також особливості їх застосування в практичних завданнях.

Основними фреймворками для побудови глибоких моделей є **PyTorch** та **TensorFlow** - обидва надають гнучкі інтерфейси для реалізації нейронних мереж, оптимізації та розподіленого навчання. Бібліотека **Hugging Face Transformers** базується на цих фреймворках та забезпечує стандартизований підхід до створення та тренування сучасних мовних моделей, включаючи GPT, BERT, та інші.

Процес навчання моделі включає:

- проєктування архітектури з нуля або на основі існуючих схем (transformers.PreTrainedModel);
- підготовку вхідних даних та **токенізацію** - перетворення тексту у формат, зручний для обробки моделлю. За це відповідають класи Tokenizer або AutoTokenizer, які враховують специфіку обраної архітектури та підтримують такі формати, як WordPiece, Byte-Pair Encoding (BPE), SentencePiece тощо;
- визначення гіперпараметрів та конфігурацій (TrainingArguments, Trainer);
- тренування моделі з використанням локальних або хмарних обчислювальних ресурсів.

Особлива увага приділяється вибору апаратної платформи для тренування:

- **CPU (центральний процесор)** - підходить лише для тестування чи невеликих моделей через низьку швидкість обробки тензорів;
- **GPU (графічний процесор)** - стандарт для навчання великих моделей, дозволяє значно прискорити обчислення за рахунок паралельної обробки даних;
- **TPU (тензорний процесор)** - спеціалізоване апаратне забезпечення для прискореного навчання та інференсу моделей, ефективне для великих обсягів даних та розподіленого навчання.

Моніторинг якості навчання виконується за допомогою **Weights & Biases (wandb)** -

платформи, яка дозволяє відстежувати метрики, втрати, точність, візуалізувати прогрес навчання, а також порівнювати експерименти між собою.

Серед основних викликів повного навчання:

- **високі вимоги до апаратного забезпечення** - трансформерні моделі можуть вимагати десятки гігабайт відеопам'яті;
- **великий обсяг навчальних даних**, які слід завантажити, почистити, токенізувати та оптимізувати;
- **тривалий час тренування**, навіть за умови використання кількох потужних GPU/TPU;
- **необхідність ретельної оптимізації** навчального процесу та забезпечення його повторюваності.

Попри складність, екосистема Python забезпечує весь цикл створення моделей: від архітектури до навчання, тестування та подальшого використання, що робить її основою більшості сучасних досліджень і продуктів у сфері штучного інтелекту.

Висновок:

Розробка та навчання моделей штучного інтелекту на сучасному етапі є комплексним процесом, який охоплює проєктування архітектури, обробку вхідних даних, токенізацію, налаштування гіперпараметрів і вибір відповідної апаратної платформи. Екосистема Python із її провідними бібліотеками – PyTorch, TensorFlow, Transformers – забезпечує ефективні засоби для реалізації повного циклу розробки моделей. Водночас, повноцінне навчання потребує значних ресурсів, знань і практичного досвіду. Впровадження систем моніторингу, таких як Weights & Biases, дозволяє значно підвищити керованість і якість тренування моделей. Загалом, використання відкритих бібліотек та відповідне програмно-апаратне забезпечення відкривають широкі можливості для наукових досліджень і розробки прикладних рішень у сфері ШІ.

Список літератури

1. Основний курс Huggingface щодо бібліотеки transformers [Електронний ресурс] https://huggingface.co/learn/llm-course/chapter1/1
2. Документація з використання Llama [Електронний ресурс] https://github.com/meta-llama/llama3
3. Документація з використання Mistral [Електронний ресурс] https://docs.mistral.ai/getting-started/quickstart/
4. Taulli T. (2024). Generative AI and a Large Language Models, In 2024 "AI-Assisted Programming" (pp 26-35)