

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЗАДАЧІ ІНТЕЛЕКТУАЛЬНОГО УПРАВЛІННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ

Швачич Г.Г.¹, Мороз Б.І.¹, Щербина П.О., Мороз Д.М.²

¹НТУ «Дніпровська політехніка», д.т.н., професор, Україна

²НТУ «Дніпровська політехніка», д.PhD., доцент, Україна

Анотація. Розглядаються основні особливості застосування мови програмування Python у сфері моделювання штучних нейронних мереж. Показано, що розвинена система бібліотек мови програмування Python значно спрощує процес розробки нейромереж. Відзначається, що однією з ключових переваг мови Python є її зручність для швидкого прототипування. В той же час інтерпретований характер мови дозволяє оперативно тестувати різні архітектури нейромереж, змінювати параметри та отримувати результати в режимі реального часу. Представлені дослідження зосереджені на спільному використанні бібліотек TensorFlow та Keras. Показано, що бібліотеки TensorFlow та Keras є потужними інструментами для моделювання та навчання нейронних мереж. Вони значно спрощують процес розробки, забезпечуючи широкий набір функцій для створення, оптимізації та впровадження моделей машинного навчання.

Ключові слова: програмне забезпечення, нейронна система, штучний інтелект, технологічний процес.

Вступ. Потреба у впровадженні сучасних систем контролю параметрів технологічних процесів є визначальним чинником стратегічного розвитку галузі, оскільки має вагоме науково-технологічне та економічне значення. У металургійному виробництві існує ряд викликів, пов'язаних із моніторингом параметрів технологічних процесів і системами їх керування. Серед основних проблем варто виділити: відхилення геометричних параметрів прокату, порушення плоскостності, поверхневі дефекти через механічні пошкодження, а також нерівномірний температурний розподіл, що спричиняє внутрішні напруження.

Сучасні підходи до вирішення цих завдань передбачають застосування розподілених систем із кластерними мережами датчиків для первинної обробки даних. Інтеграція нейронних мереж з інфрачервоними датчиками та лазерними сканерами відкриває нові перспективи для оптимізації процесів

вальцювання та прокатки металів, що сприяє зниженню виробничих витрат і покращенню якості продукції.

Водночас виникає питання ефективного використання програмного забезпечення для інтелектуального управління технологічними процесами на базі штучних нейронних мереж. Дане дослідження спрямоване на аналіз таких програмних рішень.

Основний матеріал. На сьогодні Python є однією з найпопулярніших мов програмування для моделювання та розробки нейронних мереж, що пояснюється її простотою, великою екосистемою бібліотек та активною спільнотою. Використання Python як основної мови для створення нейронних мереж зумовлене низкою факторів, зокрема зручністю моделювання, наявністю спеціалізованих бібліотек і широкими можливостями для обробки великих обсягів даних [1 – 3]. Популярність цієї мови зростає паралельно з розвитком штучного інтелекту, що робить її оптимальним вибором як для наукових досліджень, так і для комерційних рішень у сфері глибокого навчання.

Розглянемо основні особливості застосування мови програмування Python у сфері моделювання штучних нейронних мереж. Насамперед це розвинена система бібліотек для машинного навчання, яка значно спрощує процес розробки нейромереж. До найпопулярніших бібліотек належать:

- TensorFlow – одна з найпоширеніших бібліотек, що дозволяє створювати як прості, так і складні нейромережі.
- PyTorch – гнучка бібліотека, яка активно використовується у наукових дослідженнях і навчанні моделей.
- Keras – високорівнева бібліотека, що працює поверх TensorFlow і полегшує процес створення моделей.
- Scikit-learn – набір інструментів для базового машинного навчання та аналізу даних.

Однією з ключових переваг Python є його зручність для швидкого прототипування. Інтерпретований характер мови дозволяє оперативно тестувати різні архітектури нейромереж, змінювати параметри та отримувати результати в режимі реального часу.

Крім того, Python забезпечує легку інтеграцію з іншими мовами програмування та фреймворками. Завдяки модулям на кшталт Cython, Python

ефективно взаємодіє з C/C++, що дозволяє підвищити продуктивність розрахунків. Також мова добре інтегрується з популярними інструментами для обробки великих даних, такими як Apache Spark.

Ще однією важливою особливістю є широкі можливості для візуалізації та аналізу результатів. Серед найпоширеніших інструментів:

- Matplotlib, Seaborn – для створення графіків та візуалізації навчального процесу.

- TensorBoard – для моніторингу та налагодження глибоких нейронних мереж.

Завдяки цим перевагам Python залишається однією з ключових мов у сфері машинного навчання та штучного інтелекту.

Ці дослідження зосереджені на спільному використанні бібліотек TensorFlow та Keras. Зокрема, бібліотека TensorFlow є наскрізною платформою з відкритим кодом, призначеною для створення програм машинного навчання. Це символічна математична бібліотека, яка використовує потоки даних і диференційоване програмування для виконання різноманітних завдань, пов'язаних із навчанням та функціонуванням глибоких нейронних мереж. Завдяки цьому розробники можуть створювати ефективні програми машинного навчання, використовуючи широкий набір інструментів, бібліотек та ресурсів спільноти.

Процес застосування TensorFlow можна поділити на дві основні фази:

1. Фаза розвитку – етап навчання нейромережі. Навчання зазвичай відбувається на настільному комп'ютері або ноутбукі.

2. Фаза запуску (інференсу) – після завершення навчання модель може бути розгорнута на різних платформах для подальшого використання.

Важливо відзначити, що процес навчання можна виконувати на кількох машинах, а потім запускати модель на іншій пристрої. Навчання моделі можливе як на центральних (CPU), так і на графічних процесорах (GPU). Оскільки глибоке навчання передбачає виконання великої кількості операцій множення матриць, TensorFlow забезпечує їхню високу швидкість завдяки реалізації цих обчислень мовою C++. Водночас доступ до бібліотеки та керування нею здійснюється переважно через інші мови програмування, зокрема Python.

З іншого боку, Keras є високорівневою бібліотекою, яка активно використовується як у дослідницьких, так і в промислових проєктах. Саме ця властивість робить її ефективним інструментом для розробки інтелектуальних систем управління технологічними процесами на основі штучних нейронних мереж.

Приймаючи до уваги зазначене, у даних дослідженнях бібліотека Keras використовувалася для створення нейронних моделей, а TensorFlow – для їх навчання та реалізації. Важливо зазначити, що бібліотеки TensorFlow та Keras мають тісний зв'язок, оскільки Keras є високорівневою бібліотекою для глибокого навчання, яка працює поверх TensorFlow. Така інтеграція має кілька ключових переваг:

- Простота використання пояснюється тим, що бібліотека Keras забезпечує простий та інтуїтивний інтерфейс для створення нейронних мереж, тоді як бібліотека TensorFlow надає потужні інструменти для обчислень та масштабованості.

- Гнучкість реалізації процесу програмування пояснюється тим, що використовуючи бібліотеку Keras поверх бібліотеки TensorFlow, отримують доступ до всіх функцій бібліотеки TensorFlow, включаючи розподілене навчання, обробку великих даних та підтримку різних апаратних платформ.

На рис. 1 представлено код, який демонструє базові принципи використання бібліотек TensorFlow та Keras для побудови простої нейромережі.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np
```

Рисунок 1 – Особливості імпортування бібліотек для побудови нейронної мережі

Розроблена мережа функціонує наступним чином. На першому етапі здійснюється імпорт бібліотек для підключення необхідних модулів. Далі створюється модель, визначаючи архітектуру нейромережі (шари, кількість нейронів, функції активації). Наступним кроком є компіляція, де задаються алгоритм навчання (оптимізатор, функція втрат). Після цього проводиться навчання моделі, під час якого передаються вхідні дані, і мережа налаштовує

свої ваги. На завершальному етапі здійснюється прогнозування, коли модель використовується для передбачення нових значень.

Висновки. Дослідження показали, що бібліотеки TensorFlow та Keras є потужними інструментами для моделювання та навчання нейронних мереж. Вони значно спрощують процес розробки, забезпечуючи широкий набір функцій для створення, оптимізації та впровадження моделей машинного навчання.

Основні переваги сумісного використання бібліотек TensorFlow та Keras полягають у наступному:

- Зручність та гнучкість.
- Висока продуктивність.
- Масштабованість та підтримка різних платформ.
- Інтерактивність та можливість швидкого прототипування.
- Широкий набір інструментів для моніторингу та аналізу.

Отже, завдяки зазначеним широким можливостям бібліотеки TensorFlow та Keras є ідеальним вибором як для дослідників, так і для розробників комерційних рішень у сфері машинного навчання та штучного інтелекту.

ЛІТЕРАТУРА/REFERENCE

1. Shollet F. Deep Learning with Python. Manning, 2022. 469 p.
2. Raschka S., Mirjalili V. Python Machine Learning. Pask Publishing, 2019. – 770 p.
3. McKinney Wes Python for Data Analysis. O'Reilly Media. 2022. 579 p.

SOFTWARE ANALYSIS FOR THE TASK OF INTELLECTUAL CONTROL OF TECHNOLOGICAL PROCESSES

Hennadi Shvachych, Borys Moroz, Pavlo Shcherbyna, Dmytro Moroz

Abstract. *The main features of the use of the Python programming language in the field of modeling artificial neural networks are considered. It is shown that the developed system of libraries of the Python programming language significantly simplifies the process of developing neural networks. It is noted that one of the key advantages of the Python language is its convenience for rapid prototyping. At the same time, the interpreted nature of the language allows you to quickly test different architectures of neural networks, change parameters and obtain results in real time. It is noted that the Python programming language provides easy integration with other programming languages and frameworks. It is shown that the TensorFlow and Keras libraries are powerful tools for modeling and training neural networks. They greatly simplify the development process by providing a wide range of features for creating, optimizing, and deploying machine learning models.*

Keywords: *software, neural system, artificial intelligence, technological process.*