

Перелік посилань

1. Midjourney [Електронний ресурс]. – Режим доступу: https://www.midjourney.com/home_
2. Logolivery [Електронний ресурс]. – Режим доступу: https://logolivery.ai_
3. ChatGPT [Електронний ресурс]. – Режим доступу: <https://chat.openai.com>.
4. Tamkin Alex; Brundage Miles; Clark Jack; Ganguli Deep. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. 2021, 8 p.
5. Knox, W. Bradley; Stone, Peter. Augmenting Reinforcement Learning with Human Feedback. University of Texas at Austin, Department of Computer Science, 2011, 8 p.
6. О. М. Верес, Р. М. Оливко. Класифікація методів аналізу Великих даних. Вісник Національного університету "Львівська політехніка". Серія : Інформаційні системи та мережі. 2017, № 872, С.84-92.
7. Синєглазов В., Чумаченко О. Укладач Бідюк, П. І., Шугалей, Л. П. Методи та технології напівкеруваного навчання: Курс лекцій. Київ, КПІ ім. Ігоря Сікорського, 2022. С. 7-9.
8. OpenAI. ChatGPT: Optimizing Language Models for Dialogue. – 2022. – Режим доступу: <https://openai.com/blog/chatgpt>.

НЕЙРОННА МЕРЕЖА ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Соколовський А.О.

Харківський національний автомобільно-дорожній університет

Прогнозування значень часового ряду за допомогою моделі MLP є складним завданням, що вимагає систематичного та методичного підходу. Починаючи з розуміння суті задачі, важливо визначити, яка саме змінна має бути прогнозована та які фактори впливають на цю змінну в часі.

Після аналізу часового ряду та виявлення його особливостей, таких як тренди, сезонність та аномалії, наступним етапом є підготовка даних. Це може включати в себе видалення шуму, виявлення та корекцію викидів, а також розгляд можливості стандартизації або нормалізації даних.

Будівництво моделі MLP передбачає визначення її архітектури, включаючи кількість шарів, кількість нейронів у кожному шарі та вибір функцій активації.

Навчання моделі включає в себе вибір оптимального алгоритму оптимізації, розділення даних на тренувальний та валідаційний набори, а також моніторинг функції втрати під час тренування. Ефективна оцінка моделі базується на використанні відповідних метрик, таких як середньоквадратична помилка (MSE) чи середньоабсолютна помилка (MAE), та аналізі залишкових графіків.

Етап гіперпараметризації та налаштування дозволяє експериментувати з параметрами моделі, такими як швидкість навчання чи кількість епох, для досягнення оптимальної продуктивності. Важливо також використовувати

крос-валідацію для підтвердження стійкості та генералізації моделі на різних наборах даних.

Після успішного навчання модель готова до прогнозування на нових даних та впровадження в робоче середовище. Систематичний моніторинг її продуктивності та можливе оновлення відповідно до змін в умовах завдання чи в самому часовому ряду є ключовим етапом в подальшому використанні моделі.

Характеристика об'єкта дослідження

Об'єктом дослідження є датасет "Air Passenger", який охоплює історичні дані щодо пасажирських перевезень авіатранспортом. Ці відомості включають кількість пасажирів, які скористалися авіаперевезеннями протягом різних періодів часу. За допомогою цього датасету прагнемо розробити прогностичну модель, базуючись на нейронній мережі з архітектурою MLP.

Однією з ключових складових датасету є інформація про дати проведення перевезень та відповідні обсяги пасажиропотоку. Ці дані формують часовий ряд, що репрезентує динаміку змін у кількості пасажирів з плином часу. Проведення аналізу цього часового ряду є невід'ємною частиною для розуміння його структури та виявлення можливих закономірностей, таких як сезонні коливання, тренди або випадкові величини.

Головною метою є створення прогностичної моделі, яка здатна точно прогнозувати майбутні значення пасажиропотоку. Використання нейронної мережі з моделлю MLP надає можливість адаптуватися до складних закономірностей в даних та забезпечує ефективне прогнозування.

Отримані результати дослідження можуть стати важливим внеском у розуміння та оптимізацію пасажирських перевезень в авіаційній галузі. Поєднання аналізу часового ряду та застосування нейронної мережі відкриває можливості для розробки ефективних стратегій управління ресурсами та планування на основі точних прогнозів.

Реалізація створеної нейронної мережі

Для реалізації нейронної мережі в коді використовується бібліотека `scikit-learn`, а саме клас `MLPRegressor`, який надає можливість створювати та навчати штучні нейронні мережі для задач регресії. Відома також як мультишаровий перцептрон (MLP), ця модель є однією з популярних архітектур нейронних мереж. Процедура тренування штучної нейронної мережі включає ініціалізацію моделі з визначенням її архітектури, навчання моделі на тренувальних даних та використання навченої моделі для прогнозування на нових даних.

Практична реалізація відбувається завдяки використанню бібліотек та інструментів мови програмування Python, зокрема бібліотеки `scikit-learn` для роботи з машинним навчанням та нейронними мережами. Результати виводяться на графіку з використанням бібліотеки `matplotlib`.

Код реалізації розбитий на кілька етапів для зручності розробки та розуміння процесу. Основні етапи включають:

1. Завантаження та Підготовка Даних: в цьому етапі відбувається завантаження даних з CSV-файлу за допомогою бібліотеки `pandas`. Конвертація стовбця 'Month' у формат дати та встановлення його як індекс. Розділення даних на тренувальний та тестовий набори.

2. Масштабування даних: на цьому етапі використовується `MinMaxScaler` для масштабування значень у діапазон від 0 до 1.

3. Створення часових послідовностей: відбувається визначення функції `create_sequences` для створення часових послідовностей для навчання та тестування.

4. Навчання моделі MLP: Визначення та навчання моделі `MLPRegressor` з використанням даних тренувального набору.

5. Прогнозування на тестових даних: використання навченої моделі для прогнозу значень на тестовому наборі.

6. Оцінка якості моделі: розрахунок середньоквадратичної помилки (MSE) для оцінки точності прогнозів.

7. Прогнозування майбутніх даних: прогноз майбутніх значень, використовуючи останні точки тестового набору.

8. Зворотне масштабування: зворотне масштабування прогнозованих та фактичних значень для відображення на графіку.

9. Візуалізація результатів: використання `matplotlib` для створення графіка, на якому відображаються фактичні дані, прогнозовані значення на тестовому наборі та майбутні прогнози.

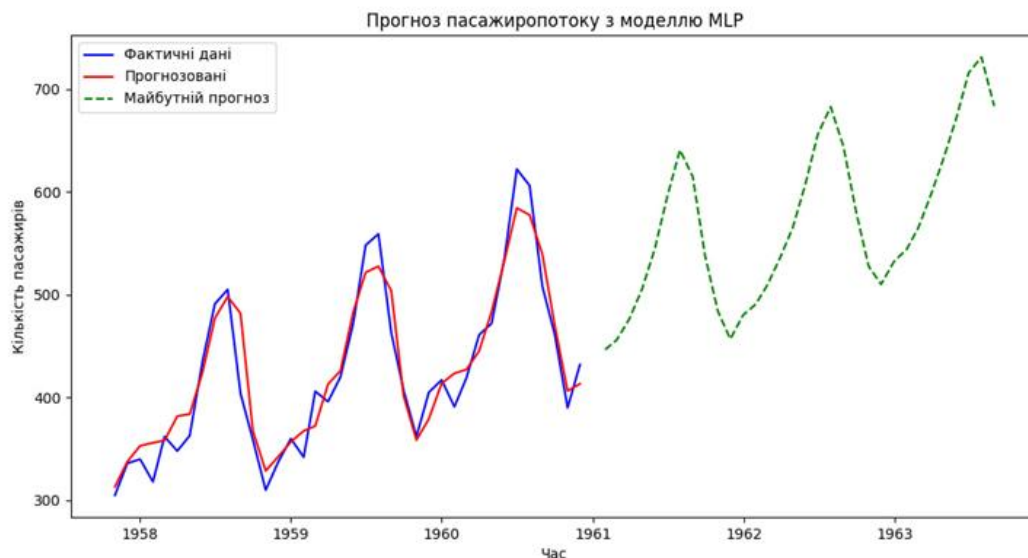


Рисунок 1 – Результат роботи програми

Під час виконання була використана штучна нейронна мережа (MLP), яка є видом глибокого навчання, для вирішення задачі прогнозування пасажиро-

поток на основі часового ряду пасажирських перевезень. Починаючи з завантаження та обробки даних за допомогою бібліотеки Pandas, дати у стовпці "Month" конвертуються у формат дати. Далі дані масштабуються до діапазону від 0 до 1 за допомогою MinMaxScaler з бібліотеки scikit-learn, що дозволяє нормалізувати дані для ефективного тренування моделі.

Цей код не лише демонструє процес прогнозування пасажиропотоку, але і надає основу для розвитку та удосконалення моделей прогнозування часових рядів з використанням нейронних мереж.

Висновок

У рамках виконаної роботи з прогнозування часового ряду пасажиропотоку використано штучну нейронну мережу (MLP). Перший етап включав підготовку та завантаження даних, де часовий ряд був оброблений та встановлено у формат datetime. Далі, для забезпечення стабільності навчання, дані були нормалізовані за допомогою MinMaxScaler.

Слідом за цим була реалізована функція для створення часових послідовностей, що допомагала враховувати залежності в часовому ряді. Нейронна мережа (MLPRegressor) була визначена та навчена на підготовлених даних, з врахуванням визначених параметрів архітектури та оптимізації.

Застосовано навчену модель для прогнозування значень на тестових даних, а якість прогнозу оцінено за допомогою середньоквадратичної помилки (MSE). Крім того, використано модель для прогнозування майбутніх значень часового ряду. Остаточні результати були візуалізовані за допомогою Matplotlib, де порівнювалися фактичні, прогнозовані та майбутні значення. Всі ці етапи взаємодіють, допомагаючи створити ефективний інструмент для прогнозування та визначення тенденцій в часовому ряді пасажиропотоку.

Отже, використання MLP дозволило створити модель, яка ефективно прогнозує значення часового ряду пасажиропотоку. Підготовка даних, правильний вибір архітектури та параметрів моделі, а також оцінка її точності через візуалізацію допомогли створити функціональний інструмент для прогнозування майбутніх значень в даному контексті.

Список джерел

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50 – P. 159–175.

Python Documentation - Scikit-learn. .: [Електронний ресурс].-Режим доступу: <https://scikit-learn.org/stable/documentation.html>

Pandas Documentation. .: [Електронний ресурс].-Режим доступу: <https://pandas.pydata.org/pandas-docs/stable/index.html>

Каллан, Р. Нейронні мережі: Короткий довідник / Р. Каллан. - М .: Вільямс І.Д., 2017. - 288 с.

Paolo Galeone Dropout — Методи рішення проблеми перенавчання в нейронних мереж [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/company/wunderfund/blog/330814/>.

J. Park, J. I.W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 1991. – 151 с.