

М. О. Шестакович¹, Ю. В. Шабатура²

¹ магістр, аспірант, кафедра інформаційних систем та комп'ютерного моделювання, Національний лісотехнічний університет України, Львів, Україна, <https://orcid.org/0009-0005-6714-1582>

² д.т.н., проф кафедри електромеханіки та електроніки, Національна академія сухопутних військ імені гетьмана П. Сагайдачного, Львів, Україна, <https://orcid.org/0000-0002-9961-1244>

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ПЕРЕХОДУ ІНФОРМАЦІЙНИХ СИСТЕМ З МОНОЛІТНОЮ АРХІТЕКТУРОЮ В МІКРОСЕРВІСНУ

Анотація. Проведено аналіз застосування нейронних мереж для оптимізації процесу переходу від монолітної архітектури до системи на основі мікросервісів. Основна увага приділяється аналізу потенціалу нейронних мереж у виявленні залежностей між компонентами системи, їх декомпозиції та прогнозуванні навантажень на окремі сервіси. Використовуючи нейронні мережі, ми можемо підвищити гнучкість, масштабованість і надійність програмних систем, забезпечуючи оптимальний розподіл функцій та ресурсів. Робота також розглядає перспективи та виклики, пов'язані з впровадженням цих технологій, включаючи занепокоєння щодо обчислювальних ресурсів та інтерпретованості моделей.

Ключові слова: нейронні мережі, мікросервіси, оптимізація, монолітна архітектура, прогнозування.

Оскільки попит на гнучкість та масштабованість програмних систем продовжує зростати, процес переходу від монолітної архітектури до архітектури на основі мікросервісів стає дедалі популярнішим. Розбиваючи великі системи на незалежні сервіси, мікросервісна архітектура дозволяє більш ефективно управляти ресурсами та спрощує процес підтримки та оновлення складних систем.

Проте цей перехід супроводжується значними викликами, такими як декомпозиція системи, організація взаємодії між мікросервісами та забезпечення узгодженості даних [1]. Одним із перспективних напрямків для вирішення цих питань є застосування нейронних мереж, які пропонують потужні інструменти для аналізу та оптимізації складних систем.

Нейронні мережі мають унікальну здатність обробляти великі обсяги даних, виявляти приховані закономірності та пропонувати інноваційні рішення. У контексті автоматизації процесу переходу їх можна використовувати для аналізу залежностей між компонентами, визначення оптимальних шляхів декомпозиції та прогнозування навантажень на мікросервіси. Використовуючи цей потенціал, можна створювати більш гнучкі, адаптивні та стійкі системи, забезпечуючи підвищену ефективність і продуктивність [2].

В останніх наукових дослідженнях вивчення шляхів оптимізації процесу переходу через інтеграцію інтелектуальних алгоритмів стало важливим кроком вперед у розвитку сучасних підходів до проектування програмного забезпечення. У міру того, як ми стикаємося з викликами у розробці програмних систем, нас все більше приваблюють інноваційні рішення, здатні спростити та оптимізувати процес, мінімізуючи вузькі місця та затримки.

У традиційному підході до монолітної архітектури всі функціональні компоненти інтегровані в єдине програмне середовище. Така архітектура забезпечує простоту розробки та тестування, оскільки всі модулі знаходяться в одному місці, а комунікація між ними відбувається з мінімальною затримкою. Однак цей підхід має значні обмеження, особливо

під час масштабування системи та внесення змін. Наприклад, при збільшенні навантаження потрібно масштабувати всю систему, а не окремі компоненти, що може призвести до зниження ефективності [3].

Мікросервісна архітектура стала відповіддю на ці виклики, надаючи можливість розбити програмне забезпечення на окремі сервіси, кожен з яких виконує конкретну бізнес-функцію. Кожен мікросервіс працює незалежно, взаємодіє з іншими через добре визначені API і може бути розгорнутий, оновлений або масштабований окремо від інших частин системи. Це значно спрощує управління системою, підвищує її гнучкість та дозволяє ефективно масштабувати окремі компоненти в залежності від потреб. Наприклад, система може продовжувати функціонувати навіть у разі відмови одного з мікросервісів, оскільки інші сервіси працюватимуть автономно [1].

Незважаючи на всі переваги мікросервісів, їх впровадження супроводжується низкою проблем. Перш за все, розподіл монолітної системи на мікросервіси є складним завданням, що потребує ретельного аналізу існуючої архітектури та визначення точок декомпозиції. Також взаємодія між мікросервісами потребує розробки надійних комунікаційних механізмів для забезпечення ефективного обміну даними між сервісами. Інтеграція мікросервісів також супроводжується складнощами в управлінні даними та забезпеченні їхньої консистентності, що вимагає використання відповідних технологій і методів [4].

Використання нейронних мереж є перспективним підходом для оптимізації цього процесу завдяки здатності цих моделей обробляти великі обсяги даних і виявляти приховані патерни, які можуть бути неочевидними для аналітиків. Моделі нейронних мереж дозволяють автоматизувати аналіз складних систем, значно знижуючи ризики при переході до нової архітектури.

Нейронні мережі можуть бути використані для визначення найкращих точок розділення, що допомагає виявити, які елементи системи можна розділити на окремі мікросервіси. Вони також можуть прогнозувати майбутні навантаження на мікросервіси та оцінювати масштабованість, що є важливим аспектом для мінімізації ризику виникнення проблемних маршрутів у новій архітектурі [2].

Використання нейронних мереж може значно спростити процес виявлення зв'язків між різними елементами монолітної системи, що є складним аспектом під час переходу до архітектури мікросервісів. Це завдання зазвичай включає глибокий аналіз усього коду та розуміння того, як компоненти взаємодіють між собою. Нейронні мережі автоматизують цей аналіз, досліджуючи код і визначаючи зв'язки між різними частинами системи. В результаті, вони допомагають створити модель, яка ілюструє взаємодію компонентів, що полегшує перехід до мікросервісної структури.

Графові нейронні мережі (GNN) добре справляються з аналізом графових структур даних, що показують взаємодію між елементами системи. Це допомагає визначити ключові компоненти для можливого об'єднання в один мікросервіс або встановлення важливих залежностей після декомпозиції [5].

Прогнозування навантаження на окремі частини системи є ключовим під час переходу від єдиної структури до мікросервісної архітектури. У таких системах кожен сервіс може вимагати різних ресурсів та масштабованості, що ускладнює підтримку стабільності системи без ефективного управління навантаженням [6]. Використання методів машинного навчання, таких як нейронні мережі, дозволяє вирішити цю проблему, аналізуючи історичні дані щодо використання ресурсів та їх розподілу між сервісами.

Рекурентні нейронні мережі включають різні типи, такі як Long Short-Term Memory (LSTM) і Gated Recurrent Unit (GRU), які є ефективними для прогнозування навантаження завдяки врахуванню попередніх станів системи. Використовуючи історичні дані про роботу системи, ці моделі можуть прогнозувати пікові навантаження, що допомагає своєчасно налаштовувати сервіси та уникати перевантаження компонентів. Такий підхід сприяє кращому управлінню ресурсами та підвищує загальну стабільність системи.

Нейронні мережі також можуть використовуватися для адаптивного масштабування мікросервісів у реальному часі, що допомагає забезпечити стабільну роботу системи під час зміни навантаження. Прогнозування на основі нейронних мереж не лише допомагає підвищити стійкість до навантажень, але й дозволяє уникнути надмірного використання ресурсів, оптимізуючи витрати [7].

У контексті мікросервісної архітектури ефективне маршрутизування запитів між окремими сервісами є ключовим фактором для забезпечення стабільної та ефективної роботи системи. Оптимізація маршрутизації передбачає зменшення часу передачі даних, балансування навантаження між сервісами та скорочення затримок у відповідях. Використання нейронних мереж для цієї мети дозволяє впроваджувати адаптивні методи, які можуть динамічно змінювати маршрутизацію запитів залежно від поточного стану системи.

Одним із способів покращення є використання глибоких нейронних мереж (DNN) для прогнозування найкращих шляхів передачі даних між мікросервісами. Нейронні мережі можуть враховувати такі фактори, як завантаженість кожного сервісу, затримки в мережі та пріоритетність запитів, що допомагає вибирати найоптимальніші маршрути для кожного запиту [8]. Це дозволяє уникнути перевантаження окремих компонентів системи та забезпечити рівномірний розподіл запитів.

Нейронні мережі також можуть використовуватися для прогнозування можливих блокувань у системі на основі аналізу історичних даних про маршрутизацію. Наприклад, коли певні мікросервіси відчують підвищене навантаження, система може змінювати маршрутизацію, перенаправляючи запити на менш завантажені сервіси. Це підвищує надійність і ефективність роботи системи в умовах змінного навантаження.

Такі підходи до оптимізації можуть також включати механізми розподілу навантаження в режимі реального часу, що дозволяє краще використовувати ресурси системи та запобігати її деградації під час пікових навантажень. Використання нейронних мереж для оптимізації переходу від монолітної до мікросервісної архітектури стало зручним інструментом для багатьох технологічних компаній, які працюють з великими даними та високими навантаженнями.

Розглянемо кілька прикладів успішного використання нейронних мереж під час переходу до мікросервісної архітектури. Яскравим прикладом є компанія Netflix [6], яка використовує інтелектуальні системи для управління своїми масштабними розподіленими сервісами. Під час переходу до мікросервісної архітектури Netflix активно використовувала рекурентні нейронні мережі (RNN) для прогнозування навантажень і оптимізації розподілу запитів між сервісами під час пікових періодів. Застосування цих підходів дозволило компанії ефективно масштабувати свої сервіси та забезпечити їх стабільну роботу навіть при значному збільшенні кількості користувачів.

Іншим прикладом є Uber [7], яка використовує нейронні мережі для негайної оптимізації своєї платформи. Компанія застосовувала згорткові нейронні мережі (CNN) для аналізу зв'язків між компонентами системи. Це дозволило виявити найбільш важливі сервіси та забезпечити ефективне розподілення ресурсів між ними, скорочуючи затримки та підвищуючи надійність платформи.

eBay [8], один з найбільших онлайн-магазинів, також перейшов до мікросервісної архітектури за допомогою нейронних мереж. Компанія використовувала автокодувальники для аналізу великих обсягів даних, кластеризації функцій та ідентифікації компонентів, які можна виділити в окремі сервіси. Це дозволило зменшити складність системи та забезпечити її гнучкість і масштабованість.

Spotify застосовувала графові нейронні мережі для аналізу взаємозв'язків між компонентами системи [9]. Це допомогло визначити найкращі шляхи розподілу функцій на мікросервіси, що зменшило затримки та підвищило продуктивність сервісу, який одночасно обслуговує мільйони користувачів.

Ці приклади демонструють, як нейронні мережі можуть допомогти вирішувати складні технічні завдання, що виникають при переході на мікросервісну архітектуру. Використання таких інструментів не тільки підвищує швидкість та надійність систем, але й забезпечує їх масштабованість у майбутньому.

Нейронні мережі мають великий потенціал у процесі переходу від монолітної до мікросервісної архітектури, проте їх використання супроводжується низкою викликів. Ці труднощі стосуються як обчислювальних ресурсів, так і складнощів інтеграції та інтерпретації результатів. Одним із головних викликів є висока обчислювальна складність нейронних мереж, особливо при роботі з великими системами. Навчання нейронних мереж вимагає значних ресурсів, таких як графічні процесори (GPU) або спеціалізовані прискорювачі, що робить їх впровадження дорогим для малих і середніх компаній. Навіть за наявності необхідних ресурсів навчання глибоких моделей може займати багато часу, особливо при аналізі великих систем з великою кількістю взаємопов'язаних компонентів.

Ще однією проблемою є "чорна скринька", коли результати роботи нейронних мереж важко інтерпретувати. Через складну структуру нейронних мереж та нелінійні зв'язки між вхідними та вихідними даними фахівцям може бути важко обґрунтувати рішення, запропоновані моделлю. Це може створювати проблеми під час прийняття управлінських рішень та впровадження змін в архітектуру системи. Успішність нейронних мереж залежить від якості даних, які використовуються для навчання моделі. Недостатні або некоректні дані можуть знизити точність прогнозів, що призведе до неправильних рішень щодо декомпозиції системи на мікросервіси [4].

Однак, попри ці проблеми, нейронні мережі мають значні перспективи в оптимізації переходу до мікросервісної архітектури. З розвитком обчислювальних технологій, зокрема зростанням потужності GPU та впровадженням нових архітектур, таких як TPU (Tensor Processing Units), час навчання нейронних мереж суттєво зменшився, що відкриває нові можливості для їхнього використання у великих масштабах. Крім того, розвиток методів пояснюваного штучного інтелекту (Explainable AI) дозволяє краще інтерпретувати результати роботи нейронних мереж, що підвищує довіру до них і полегшує їхнє впровадження у реальні бізнес-процеси.

Інтеграція нейронних мереж з іншими технологіями, такими як еволюційні алгоритми або методи оптимізації, може забезпечити більш ефективні підходи до управління складними системами. Крім того, використання нейронних мереж у поєднанні з технологіями великих даних (Big Data) відкриває нові можливості для точнішого прогнозування і більш ефективного управління ресурсами у розподілених системах. Таким чином, попри наявні виклики, нейронні мережі залишаються перспективним інструментом для оптимізації переходу з монолітної до мікросервісної архітектури, і їх розвиток продовжує відкривати нові можливості для програмного забезпечення та систем із великим навантаженням.

1. Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media. 1st Edition, 280 p. URL: <https://www.oreilly.com/library/view/building-microservices/9781491950340/>
2. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
3. Lewis, J., & Fowler, M. (2014). *Microservices: A definition of this new architectural term*. MartinFowler.com, Vol. 25, 14–26. URL: <https://martinfowler.com/articles/microservices.html>
4. Zimmermann, O. (2017). *Microservices tenets*. *Comput Sci Res Dev*, 32, 301–310. <https://doi.org/10.1007/s00450-016-0337-0>
5. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 61-80.
6. Medel, V. (2017). *Using Machine Learning and Big Data to Process Large-Scale Distributed Services at Netflix*. *IEEE International Conference on Big Data (Big Data)*.

7. Adam Gluck. (2020). *Introducing Domain-Oriented Microservice Architecture*. <https://www.uber.com/en-UA/blog/microservice-architecture/>
8. Alex Liang. (2019). *How eBay Governs its Big Data Fabric*. <https://innovation.ebayinc.com/tech/engineering/how-ebay-governs-its-big-data-fabric/>
9. Kevin Goldsmith. (2016). *Microservices at Spotify*. https://gotocon.com/dl/goto-berlin-2015/slides/KevinGoldsmith_MicroservicesSpotify.pdf