

Технічні науки

УДК 004.8

Суханюк Іван Сергійович

студент

*Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»*

Sukhaniuk Ivan

Student of the

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

Івасенко Дмитро Віталійович

студент

Національної академії внутрішніх справ

Ivasenko Dmytro

Student of the

National Academy of Internal Affairs

Потапова Катерина Романівна

кандидат технічних наук, доцент

*Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»*

Potapova Kateryna

Candidate in Technical Sciences, Associate Professor

National Technical University of Ukraine

"Igor Sikorsky Kyiv Polytechnic Institute"

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ ТЕКСТІВ

THE USE OF NEURAL NETWORKS FOR TEXT ANALYSIS

Анотація. У даному дослідженні проаналізовано різноманітні типи нейронних мереж, спеціально розроблених для аналізу текстових даних. Нейронні мережі – це потужні комп'ютерні моделі, які можуть бути навчені на великих обсягах даних і здатні розпізнавати складні залежності та патерни. Застосування нейронних мереж у галузі обробки природної мови є особливо перспективним, оскільки вона стикається із складними викликами, які вимагають більш глибокого розуміння мовного контексту. Такий підхід до аналізу тексту має широкий спектр застосувань. Зокрема, він може бути використаний для автоматичного виділення ключової інформації в новинах, сортування та категоризації документів, аналізу соціальних медіа та інших важливих сфер. Його потужний потенціал із розв'язання різноманітних завдань аналізу текстових даних підтверджує значущість та перспективність використання нейронних мереж у сучасному науковому та технологічному контексті.

Ключові слова: обробка природної мови, аналіз текстів нейронними мережами, рекурентні нейронні мережі, згорткові нейронні мережі, трансформери.

Summary. This scientific research explores various types of neural networks specifically designed for text analysis. Neural networks are powerful computer models that can be trained on large datasets and recognize complex dependencies and patterns. Their application in natural language processing holds great promise, as this field encounters challenging tasks that require deeper understanding of language context. This approach to text analysis has a wide range of applications. It can be used for automatic extraction of key information from news, document sorting and categorization, social media analysis, and many other important domains. Its powerful potential in tackling diverse text analysis tasks underscores the significance and prospects of using neural networks in the modern scientific and technological landscape.

Key words: *natural language processing (NLP), analysis of texts with neural networks, recurrent neural networks (RNN), convolutional neural networks (CNN), transformers.*

Вступ. Аналіз текстів є невід'ємною частиною сучасного інформаційного віку, де великі обсяги даних швидко накопичуються і вимагають ефективних методів обробки та розуміння. Один з ключових завдань аналізу текстів полягає у знаходженні важливих змістових елементів за допомогою ключових слів. Цей процес може бути трудомістким та часозатратним, але завдяки розширеному застосуванню нейронних мереж у сфері обробки природної мови, з'явилися нові перспективи для автоматизації цього процесу.

У даній роботі будуть представлені типи нейронних мереж для пошуку значень по ключових словах в тексті. Нейронні мережі - це комп'ютерні моделі, що навчаються на великих обсягах даних та здатні розпізнавати складні залежності та патерни. Вони широко використовуються в області обробки природної мови, де задачі аналізу текстів стають більш складними і вимагають глибокого розуміння мови.

Застосування нейронних мереж для пошуку значень по ключових словах в тексті має широкий спектр можливих застосувань. Наприклад, цей підхід може бути використаний для автоматичної ідентифікації ключової інформації в новинах, сортування та категоризації документів, аналізу соціальних медіа та багатьох інших сфер.

Матеріали та методи

Рекурентні нейронні мережі (RNN) є одним з ключових типів нейронних мереж, що використовуються для обробки послідовних даних, таких як текст. Цей клас НМ призначений для роботи з послідовностями даних, де кожен елемент послідовності впливає на обробку наступного елемента. Основний принцип роботи полягає в тому, що вона має внутрішні

зв'язки, які дозволяють передавати інформацію з попередніх станів до поточного стану мережі. Ця властивість робить RNN особливо корисними для моделювання контексту та залежностей між словами у тексті [5, с. 3; 4, с. 335].

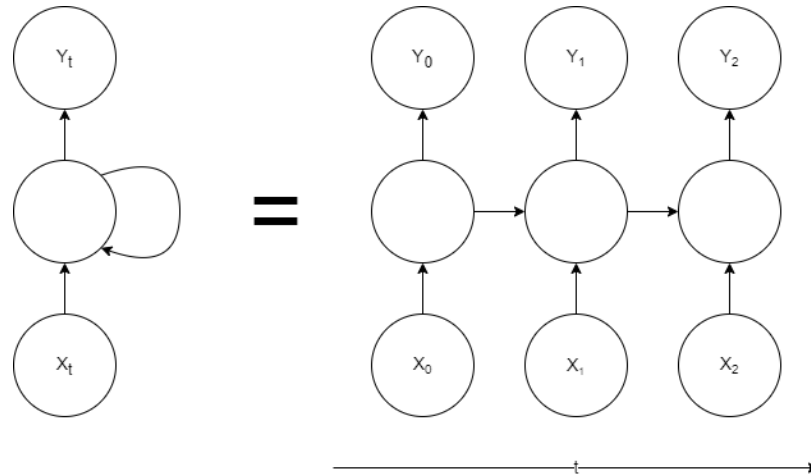


Рис. 1. Схематичне зображення RNN

Джерело: складено автором на основі [1].

Розглянемо принцип роботи RNN більш детально:

1. Послідовність вхідних даних: RNN приймає вхідні дані, які можуть бути представлені у вигляді послідовності векторів або послідовності слів. Наприклад, у випадку обробки тексту, кожне слово може бути векторизовано або закодовано у векторну форму.

2. Часові кроки: RNN обробляє вхідні дані по чергово на кожному часовому кроці. Кожен крок відповідає обробці одного елемента з послідовності.

3. Внутрішній стан: RNN має внутрішній стан, який може бути розглянутий як пам'ять моделі. Цей стан зберігає контекстуальну інформацію, отриману з попередніх кроків обробки.

4. Рекурентні зв'язки: Одна з ключових особливостей RNN - наявність рекурентних зв'язків. Кожен рекурентний шар має ваги, які зв'язують вихід на попередньому часовому кроці з входом на поточному

часовому кроці. Це дозволяє передавати інформацію з попередніх станів і використовувати її для обробки поточного вхідного елемента.

5. Функція активації: На кожному часовому кроці RNN застосовує функцію активації до комбінації ваг та вхідних сигналів, що визначають вихідний сигнал на поточному кроці. Популярні функції активації включають сигмоїду, гіперболічний тангенс та ReLU (Rectified Linear Unit).

6. Повторення процесу: Процес обробки повторюється для кожного елемента з послідовності, зв'язуючи інформацію з попередніх станів та використовуючи її для розуміння поточного вхідного елемента.

7. Вихідний результат: Після обробки всіх елементів з послідовності RNN генерує вихідний результат, який може бути використаний для подальших завдань, таких як класифікація, генерація тексту або прогнозування.

RNN здатні зберігати інформацію про попередні слова та використовувати її для аналізу поточного вхідного елемента. Таким чином, RNN можуть урахувати попередній контекст при розумінні поточного слова або фрази. Крім того, RNN можуть добре працювати з текстовими документами різної довжини, включаючи короткі та довгі тексти.

Існують такі архітектури RNN:

1. Рекурсивна нейронна мережа (Recursive RNN). Використовується для обробки ієрархічних або деревоподібних структур даних. Ця архітектура RNN здатна моделювати залежності та взаємозв'язки між елементами в вище згаданих структурах, використовуючи рекурсивну обробку та зв'язки між батьківськими та дочірніми елементами. Рекурсивні нейронні мережі використовуються для обробки структурованих даних, таких як мова, дерева синтаксичного розбору, зображення тощо [2].

2. Мережа Хопфілда. Використовується для зберігання та відновлення певних патернів пам'яті. Вона має здатність зберігати та асоціювати вхідні та вихідні шаблони з допомогою рекурентних зв'язків.

Мережа Хопфілда використовується для розпізнавання образів та вирішення оптимізаційних задач [2].

3. Мережі Елмана та Джордана. Відрізняються від звичайної рекурентної нейронної мережі (Simple RNN) у способі передачі інформації: мережа Елмана передає внутрішні стани з одного часового кроку до наступного, тоді як мережа Джордана передає вихід шару на вхідний шар. Обидва типи мереж використовують зворотні зв'язки для передачі інформації з попередніх станів до поточного стану мережі. Мережа Елмана використовується для моделювання динамічних залежностей в послідовних даних, тоді як мережа Джордана може бути використана для моделювання залежностей зміни стану або послідовності [2].

4. Мережа відлуння стану (ESN). Це тип нейронної мережі, де велика кількість випадково згенерованих ваг між нейронами залишаються незмінними під час навчання. Це дозволяє ESN ефективно обробляти послідовні дані, такі як часові ряди. Вхідні дані проходять через резервуар нейронів, який зберігає контекстуальну інформацію з попередніх часових кроків. Вихідний шар обробляє цю інформацію для прогнозування, апроксимації або класифікації [2].

5. Long Short-Term Memory (LSTM). Є розширенням рекурентних нейронних мереж (RNN), спеціально розробленим для моделювання довгострокових залежностей в послідовних даних. Вона вирішує проблему зникнення та вибуху градієнту, що можуть виникати в традиційних RNN, і здатна зберігати та використовувати інформацію залежно від її значущості. LSTM широко використовується для моделювання мови, прогнозування часових рядів, генерації тексту та інших задач обробки послідовних даних завдяки своїй здатності зберігати довгострокові залежності та ефективно обробляти складні послідовності [2; 4, с. 382; 5, с. 3-4].

6. Gated Recurrent Unit (GRU). Є спрощеною версією рекурентної нейронної мережі (RNN), що використовується для моделювання

послідовних даних. Вона має вентиля оновлення (update gate) та вихідний венти́ль (output gate), що контролюють потік інформації всередині мережі. GRU дозволяє зберігати короткострокову та довгострокову інформацію та забезпечує швидку обробку послідовностей, таких як машинний переклад та генерація тексту. Ця архітектура особливо корисна в умовах недостатку даних і в обчислювально обмежених середовищах, завдяки своїй ефективності та швидкості тренування [4, с. 389; 5, с. 4].

7. Бірекурентні мережі (Bidirectional RNN). Цей тип використовує дві незалежні рекурентні нейронні мережі, які працюють у прямому та зворотному напрямках, для обробки послідовних даних. Це дозволяє моделі враховувати контекстуальну інформацію з обох кінців послідовності. Бірекурентні мережі широко використовуються в завданнях, де важлива взаємодія між елементами послідовності, таких як машинний переклад, розпізнавання мови та аналіз тексту. Вони дозволяють здійснювати ефективну обробку і моделювання послідовних даних з урахуванням контексту з обох сторін [4, с. 399].

Згорткові нейронні мережі (CNN) є іншим типом нейронних мереж, що широко використовуються в обробці текстів. Вони спочатку були розроблені для задач обробки зображень, але виявилися ефективними й для обробки текстових даних. CNN використовують згорткові шари, пулінг та повно зв'язані шари для виявлення та використання локальних залежностей у вхідних даних. Шляхом векторизації текстових даних та застосування згорткових шарів до послідовностей слів, вони можуть виявляти локальні залежності та важливі ознаки в тексті, що робить їх потужним інструментом для класифікації тексту, визначення настрою, генерації тексту та інших задач аналізу тексту.

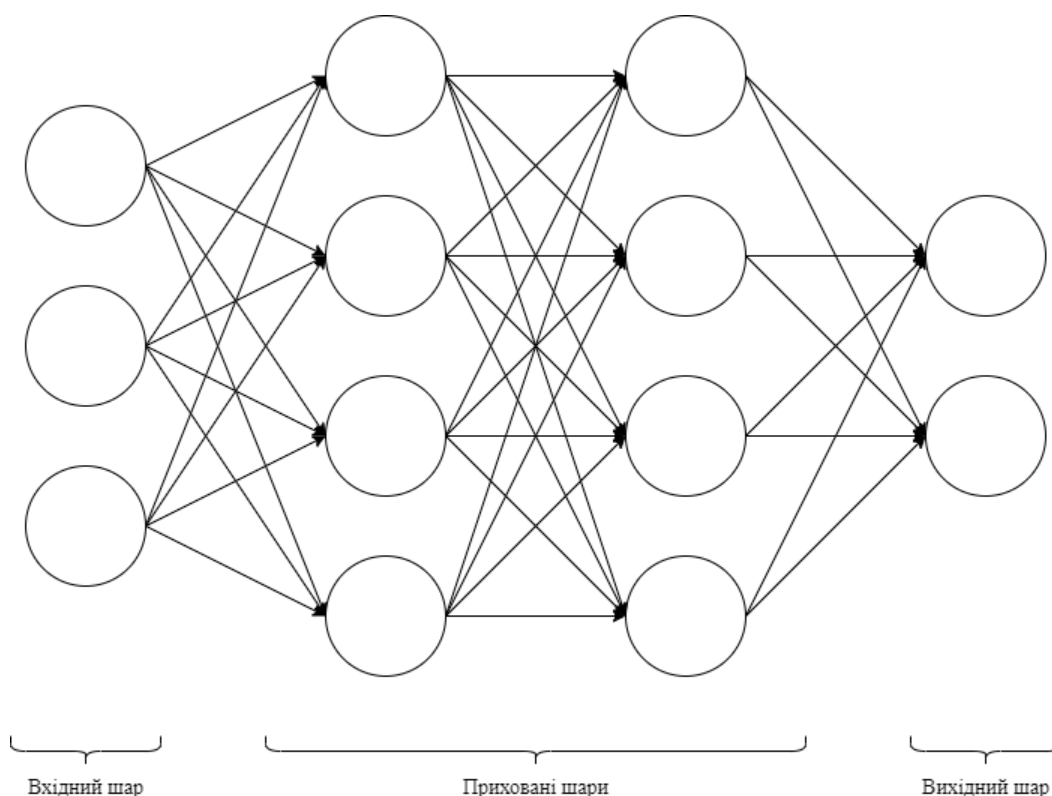


Рис. 2. Схематичне зображення CNN

Джерело: розробка автора

Розглянемо принципи роботи CNN більш детально:

1. Векторизація тексту: Спочатку текстові дані перетворюються на числові вектори, які можуть бути подані у вигляді матриці. Це можна зробити за допомогою методів, таких як однозначне кодування слів (one-hot encoding) або використання векторних моделей слів, таких як Word2Vec або GloVe. Цей крок дозволяє представити слова або символи у векторному просторі.

2. Вбудований шар (Embedding Layer): Далі, векторизовані слова передаються через вбудований шар, де кожен словний вектор замінюється на його вбудовану представлення. Цей шар навчається під час тренування мережі та дозволяє виразити семантичні відношення між словами у векторному просторі.

3. Згортковий шар (Convolutional Layer): Згорткові шари виконують операцію згортки над векторизованим текстом, аналогічно до зображень. Фільтри рухаються вздовж послідовності слів, вираховуючи скалярний добуток з підрозділами тексту. Це дозволяє виявити локальні залежності між сусідніми словами та виконати фільтрацію ознак у тексті.

4. Функція активації: Після згортки до результату застосовується функція активації, така як ReLU (Rectified Linear Unit) або sigmoid, для впровадження нелінійності в мережу.

5. Пулінговий шар (Pooling Layer): Після згортки може бути використаний пулінговий шар, який допомагає зменшити розмірність отриманих карт ознак. Популярні види пулінгу включають максимальне пулінгу (max pooling) або середнє пулінгу (average pooling), які вибирають найбільші або середні значення з підрозділів тексту.

6. Повно зв'язаний шар (Fully Connected Layer): На виході згорткових та пулінгових шарів можуть розміщуватися повно зв'язані шари, які приймають векторизовані ознаки та виконують класифікацію, регресію або інші завдання аналізу тексту.

7. Вихідний шар (Output Layer): Останній шар визначає формат виходу з мережі, залежно від конкретного завдання. Наприклад, для класифікації тексту може використовуватися шар з функцією активації Softmax, яка нормалізує вихідні значення у формі ймовірностей.

Згорткові нейронні мережі для аналізу тексту можуть використовуватися для класифікації тексту, виявлення настрою, генерації тексту та інших задач обробки тексту. Вони дозволяють автоматично виявляти важливі ознаки та залежності у текстових даних та робити складні аналізи, що робить їх потужним інструментом у сфері аналізу тексту.

Ось кілька популярних нейронних мереж для аналізу тексту:

1. TextCNN (Convolutional Neural Network for Text): TextCNN є однією з базових архітектур для аналізу тексту. Вона складається з одного

або кількох згорткових шарів, за якими слідує пулінговий шар та повно зв'язаний шар для класифікації. TextCNN використовує різні розміри фільтрів, що дозволяє виявляти ознаки різної довжини у тексті [6].

2. KimCNN: Архітектура KimCNN розширює TextCNN, додаючи кілька паралельних згорткових шарів з різними розмірами фільтрів. Це дозволяє виявляти ознаки різної довжини і підвищує точність класифікації тексту [7].

3. Hierarchical Convolutional Neural Network (HCNN): HCNN використовує ієрархічну структуру для аналізу тексту. Вона складається з двох рівнів згорткових шарів, перший рівень аналізує слова у кожному реченні, а другий рівень аналізує речення в контексті всього тексту. Це дозволяє згортковій нейронній мережі виявляти глобальні та локальні залежності у тексті [8].

4. Recurrent Convolutional Neural Network (RCNN): RCNN комбінує згорткові шари з рекурентними шарами для аналізу тексту. Вона використовує згорткові шари для виділення ознак з локальних контекстів, а потім рекурентний шар для узагальнення цих ознак у контексті всього тексту. Це дозволяє враховувати довгострокові залежності та контекст у тексті [9].

5. Transformer-based models: Останнім часом моделі на основі Transformer, такі як BERT (Bidirectional Encoder Representations from Transformers) та GPT (Generative Pre-trained Transformer), стали домінуючими в аналізі тексту. Вони використовують self-attention механізми для виявлення залежностей між словами в тексті та досягають вражаючих результатів у завданнях обробки мови [10].

Трансформери є потужним типом нейронних мереж, які були вперше запропоновані для завдань машинного перекладу, але широко застосовуються і для аналізу тексту, включаючи пошук значень по ключовим словам. Основна ідея трансформерів полягає в увазі на важливість контексту

для розуміння слів у тексті. Вони використовують механізми уваги для врахування взаємодії між словами в тексті та надання ваги різним частинам вхідної послідовності. Трансформери не мають рекурентних або згорткових шарів, але використовують стеки самоувагових шарів, які дозволяють моделі зосередитися на різних аспектах вхідного тексту одночасно. Один з ключових компонентів трансформера - це механізм уваги, який дозволяє моделі визначати, які частини тексту більш важливі для розуміння контексту. Він дозволяє моделі приділяти більше уваги важливим словам або фразам у тексті. Механізм уваги дозволяє моделі виявляти залежності між словами на різних відстанях, що робить його особливо корисним для розуміння контексту та виявлення значень ключових слів.

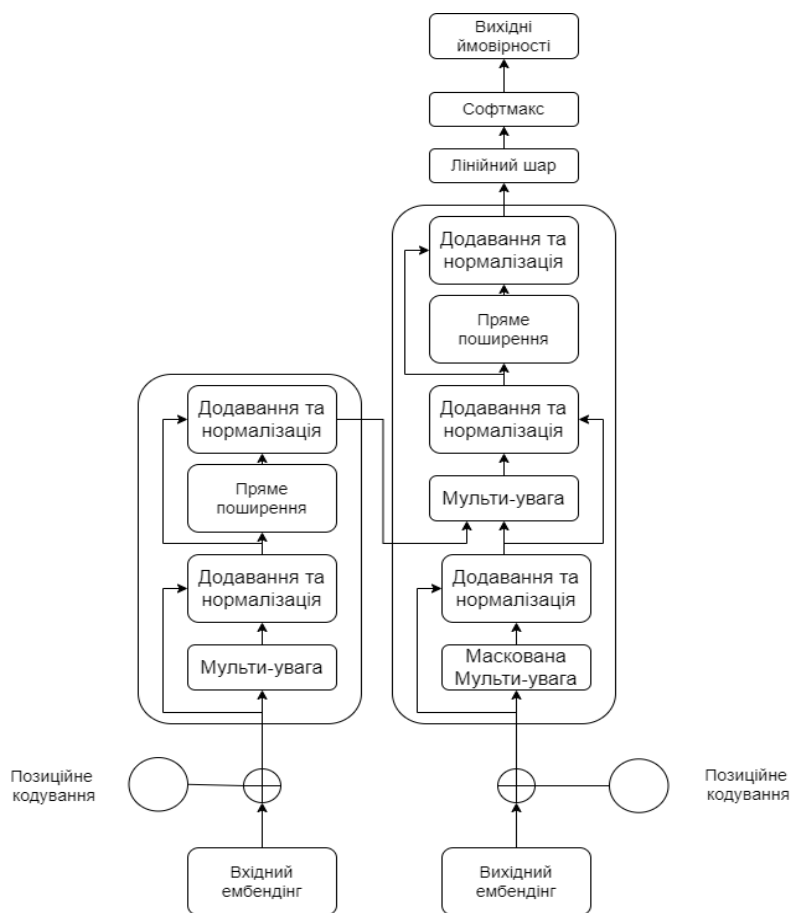


Рис. 3. Схематичне зображення архітектури трансформера

Джерело: розроблено автором на основі [12, с. 3].

Розглянемо основні принципи роботи трансформерів детальніше:

1. Вхідні ембедінги: Спочатку вхідна послідовність даних (наприклад, речення або послідовність слів) перетворюється в числові вектори, відомі як ембедінги. Цей етап може включати токенізацію, векторизацію та нормалізацію даних.

2. Енкодер: Вхідні ембедінги проходять через стек енкодерів. Кожен енкодер включає два основних компоненти: механізм уваги та позиційно-залежну фідуючу нейронну мережу (position-wise feed-forward neural network).

а) Механізм уваги: Механізм уваги дозволяє моделі визначати важливість кожного елемента вхідної послідовності залежно від контексту. Він працює наступним чином: для кожного елемента вхідної послідовності, називаємого запитом, обчислюються ваги уваги, які вказують на важливість інших елементів, називаємої ключовою послідовністю. Потім використовуються ці ваги, щоб обчислити зважену суму значень ключової послідовності, яка слугує вагованим контекстом для запиту. Цей процес повторюється для кожного елемента вхідної послідовності, що дозволяє моделі враховувати важливу інформацію з усієї послідовності.

б) Позиційно-залежна фідуюча нейронна мережа: Після механізму уваги застосовується позиційно-залежна фідуюча нейронна мережа. Це простий шаровий блок, який застосовується до кожного елемента послідовності окремо. Він допомагає моделі здатність до нелінійного перетворення та зберігати локальну інформацію.

3. Декодер: Після проходження через енкодер вхідні ембедінги проходять через стек декодерів. Кожен декодер також включає механізм уваги та позиційно-залежну фідуючу нейронну мережу. Однак, декодери також отримують додатковий сигнал уваги з енкодера. Це допомагає декодеру фокусуватись на релевантній інформації з енкодера під час генерації вихідної послідовності.

4. Генерація вихідної послідовності: На останньому шарі декодера застосовується повнозв'язний шар з функцією активації Softmax, який відображає вихідні значення у ймовірності для кожного елемента вихідної послідовності. Таким чином, модель генерує ймовірнісний розподіл для кожного елемента вихідної послідовності.

5. Тренування та оптимізація: Трансформерні мережі тренуються з використанням задачі навчання з учителем, де вихідна послідовність порівнюється з правильними вихідними значеннями для обчислення втрат. Градієнти обчислюються відносно цих втрат і використовуються для підсилення моделі за допомогою алгоритмів оптимізації, таких як стохастичний градієнтний спуск.

Таким чином, трансформерна нейронна мережа використовує механізм уваги для моделювання взаємодії між елементами вхідних послідовностей. Ця архітектура дозволяє моделювати довготривалі залежності та здатність до нелінійного перетворення даних. Трансформери зазнали значного успіху в області обробки природної мови та інших задач, де послідовність даних важлива.

Ось кілька популярних нейронних мереж для аналізу тексту:

1. Transformer: Оригінальна архітектура трансформера, яка використовує механізм уваги для моделювання взаємодії між елементами послідовності. Складається з енкодера та декодера для обробки вхідних та вихідних послідовностей відповідно [12].

2. BERT (Bidirectional Encoder Representations from Transformers): Архітектура, яка попередньо навчається на великому обсязі тексту та використовує маскуванню уваги та двобічне моделювання для засвоєння широких контекстуальних залежностей між словами [13].

3. GPT (Generative Pre-trained Transformer): Авторегресивна модель трансформера, яка спеціалізується на генерації тексту. Попередньо

навчається на великому обсязі тексту для засвоєння статистики мови та синтаксичних залежностей [14].

4. XLNet: Розширена версія BERT, яка використовує перестановки маскування уваги для моделювання залежностей між всіма елементами вхідної послідовності. Це дозволяє гнучкіше захоплювати залежності та змінювати порядок аналізу елементів [15].

5. RoBERTa (Robustly Optimized BERT Pretraining Approach): Оптимізована версія BERT, яка включає покращені методи попереднього навчання та навчання на більшому обсязі даних. Досягає кращої точності та здатності до узагальнення порівняно з BERT [16].

6. DistilBERT: Легка версія BERT, яка має меншу кількість параметрів, що дозволяє їй бути швидшою та менш вимогливою до обчислень, приблизно не втрачаючи точність порівняно з BERT [17].

Результати та висновки. Взагалі, застосування нейронних мереж для текстів є перспективним напрямком досліджень. Дослідження в цій області може привести до розробки нових алгоритмів та підходів, що покращать точність і швидкість аналізу значень у текстових документах.

На основі проведеного огляду можна зробити висновок, що нейронні мережі виявляються потужним інструментом для аналізу текстів та пошуку значень по ключовим словам. Вибір конкретного типу нейронної мережі залежатиме від особливостей конкретної задачі та вхідних даних.

Подальші дослідження в цій області можуть спрямовуватися на покращення ефективності та точності пошуку, врахування багатомовності та вивчення розширених методів аналізу тексту, що приведе до нових інноваційних рішень у цій сфері.

Література

1. What are recurrent neural networks? // Офіційний сайт компанії IBM.
URL: <https://www.ibm.com/topics/recurrent-neural-networks> (дата звернення: 10.07.2023)
2. Нгуен А.В., Сидоров Я. Е. Рекурентна нейронна мережа для обробки великих текстових даних. Системи управління, навігації та зв'язку. Збірник наукових праць. Полтава : ПНТУ, 2018. Т. 4 (50). С. 135-138.
URL: <http://journals.nupp.edu.ua/sunz/article/view/1214/1022> (дата звернення: 10.07.2023)
3. Онищенко К. Г., Данієль Я., Каменєв Р. Аналіз методів обробки природної мови. Тези доповіді 9-ої МНТК «Інформаційні системи та технології ICT-2020». 2020. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/a26452b9-d866-4aaa-aa04-524a9cae5f55/content> (дата звернення: 09.07.2023)
4. Zhang A., Lipton Z. C., Li M., Smola A. J. Dive into Deep Learning. URL: <https://d2l.ai/d2l-en.pdf> (дата звернення: 12.07.2023)
5. Wang X., Jiang W., Luo Zh. Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan, December 11-17, 2016. P. 2428–2437. URL: <https://aclanthology.org/C16-1229.pdf> (дата звернення: 12.07.2023)
6. Gong L., Ji R. What Does a TextCNN Learn? 2018. URL: <https://arxiv.org/pdf/1801.06287.pdf> (дата звернення: 13.07.2023)
7. Kim Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar. October 25-29, 2014. P. 1746–1751. URL: <https://aclanthology.org/D14-1181.pdf> (дата звернення: 13.07.2023)

8. Gao Sh., Ramanathan A., Tourassi G. Hierarchical Convolutional Attention Networks for Text Classification. Proceedings of the 3rd Workshop on Representation Learning for NLP. Melbourne, Australia, July 20, 2018. P. 11–23. URL: <https://aclanthology.org/W18-3002.pdf> (дата звернення: 13.07.2023)
9. Lai S., Xu L., Liu K., Zhao J. Recurrent Convolutional Neural Networks for Text Classification. Proceedings of the AAAI Conference on Artificial Intelligence. 2015. 29(1). URL: <https://ojs.aaai.org/index.php/AAAI/article/view/9513> (дата звернення: 13.07.2023)
10. Liu Yu., Sun G., Qiu Y., Zhang L., Chhatkuli A., Gool L. V. Transformer in Convolutional Neural Networks. 2021. URL: <https://homes.esat.kuleuven.be/~konijn/publications/2021/Liu2.pdf> (дата звернення: 13.07.2023)
11. What are convolutional neural networks? // Офіційний сайт компанії IBM. URL: <https://www.ibm.com/topics/convolutional-neural-networks> (дата звернення: 15.07.2023)
12. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. 31st Conference on Neural Information Processing Systems. Long Beach, CA, USA. 2017. URL: <https://arxiv.org/pdf/1706.03762.pdf> (дата звернення: 17.07.2023)
13. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. URL: <https://arxiv.org/abs/1810.04805> (дата звернення: 17.07.2023)
14. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language Models are Unsupervised Multitask Learners. 2019. URL: <https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf> (дата звернення: 17.07.2023)

15. Yang Zh., Dai Z., Yang Y., Carbonell J., Salakhutdinov R., Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. 2020. URL: <https://arxiv.org/pdf/1906.08237.pdf> (дата звернення: 17.07.2023)
16. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019. URL: <https://arxiv.org/pdf/1907.11692.pdf> (дата звернення: 17.07.2023)
17. Sanh V., Debut L., Chaumond J., Wolf Th. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2020. URL: <https://arxiv.org/pdf/1910.01108.pdf> (дата звернення: 17.07.2023)