Електронний журнал «Ефективна економіка» включено до переліку наукових фахових видань України з питань економіки (Категорія «Б», Наказ Міністерства освіти і науки України № 975 від 11.07.2019). Спеціальності – 051, 071, 072, 073, 075, 076, 292.

Ефективна економіка. 2024. № 2.

DOI: http://doi.org/10.32702/2307-2105.2024.2.80

УДК: 004.91

Т. Л. Кмитюк,
к. е. н., доцент,
доцент кафедри математичного моделювання та статистики,
Київський національний економічний університет імені Вадима Гетьмана
ORCID ID: https://orcid.org/0000-0001-5262-856X
А. А. Завальський,
магістрант спеціальності «Економічна кібернетика та Дата Сайанс»,
Київський національний економічний університет імені Вадима Гетьмана
ORCID ID: https://orcid.org/0009-0008-1113-2152

АНАЛІЗ МОДЕЛЕЙ ГЛИБОКОГО ТА МАШИННОГО НАВЧАННЯ ДЛЯ РОБОТИ З ПРИРОДНОЮ МОВОЮ

T. Kmytiuk,
PhD in Economics,
Associate Professor of the Department of Mathematical Modeling and Statistics,
Kyiv National Economic University named after Vadym Hetman
A. Zavalskyi,
Master's student of "Economic cybernetics and Data science", Kyiv National Economic University named after Vadym Hetman

ANALYSIS OF DEEP LEARNING AND MACHINE LEARNING MODELS FOR NATURAL LANGUAGE PROCESSING

У зв'язку зі стрімким розвитком машинного та глибокого навчання та його застосування в обробці природної мови (Natural Language Processing, NLP), виникає низка проблем та викликів, які необхідно детально проаналізувати. Ця стаття спрямована на розгляд існуючих моделей глибокого навчання для роботи з природною мовою, виявлення їхніх переваг та обмежень, а також визначення ключових напрямків для подальшого вдосконалення та використання в різних галузях. Починаючи з традиційних методів, таких як Bag-of-Words та TF-IDF, стаття переходить до огляду більш сучасних архітектур, зокрема згорткових та рекурентних нейронних мереж. Стаття оцінює ефективність цих моделей в різноманітних завданнях NLP, включаючи визначення тону тексту, відповіді на питання та машинний переклад. Висвітлюються важливі аспекти, такі як якість та обсяг навчальних даних, швидкість навчання та витрати на обчислення.

Due to the rapid development of machine learning and deep learning, as well as their application in Natural Language Processing (NLP), a series of problems and challenges arise that need to be thoroughly analyzed. The article provides a comprehensive examination of contemporary approaches in Natural Language Processing (NLP) through deep learning and machine learning models. Beginning with a review of traditional methods like Bag-of-Words and TF-IDF, the article explores their foundational role while acknowledging their limitations in capturing semantic relationships. The focus then shifts to the transformative impact of Deep Neural Networks (DNNs), specifically Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), in capturing local and sequential dependencies within texts.

The article assesses the effectiveness of these models in various NLP tasks, including sentiment analysis, question answering, and machine translation.

Important aspects such as the quality and quantity of training data, training speed, and computational costs are highlighted.

However, the article emphasizes that model accuracy and efficiency depend not only on architecture but also on the quality and quantity of training data. The challenges of overfitting or under fitting due to insufficient data are highlighted, underscoring the importance of robust datasets.

Moreover, considerations such as learning speed, training and inference costs, and scalability are discussed, emphasizing the need for selecting models aligning with project-specific requirements and constraints.

Looking forward, the article suggests future research directions, anticipating further refinement of architectures, the development of novel approaches for specific tasks, and a focus on efficient training and utilization methods for NLP models. Overall, this analysis provides valuable insights into the evolving landscape of NLP, offering researchers and practitioners a nuanced understanding of the strengths, limitations, and future prospects in the realm of deep learning and machine learning models for natural language processing.

Ключові слова: *природна мова, класифікація текстів, Bag-of-Words, частота термінів – інверсна частота документів, машинне та глибоке навчання, нейронні мережі.*

Keywords: *natural language, text classification, Bag-of-Words, term frequency - inverse document frequency, machine and deep learning, neural networks.*

Постановка проблеми. У сучасному цифровому світі, де інформація є ключовим ресурсом, велика увага приділяється розвитку технологій обробки природної мови (Natural Language Processing, NLP) за допомогою моделей

глибокого та машинного навчання. Глибоке навчання в сфері обробки природної мови використовується для розв'язання різноманітних задач, таких як машинний переклад, розпізнавання мовлення, сентимент-аналіз, аналіз тональності, витягування інформації та багато інших. Однією з ключових проблем є відсутність універсальної моделі глибокого навчання, яка б задовольняла всі потреби в обробці природної мови. Різні завдання, такі як машинний переклад, аналіз настроїв, та генерація контенту, вимагають різних підходів, існуючі моделі можуть бути недостатньо ефективними для всіх сценаріїв використання.

Аналіз останніх досліджень і публікацій. Інструментарій та техніка глибокого та машинного навчання для обробки природної мови (Natural Language Processing, NLP) є активним предметом досліджень вченими з усього світу. Традиційні методи розглядаються в роботах Q. Wisam та ін. [1], C. Sammut та G.I. Webb [2], T. Georgieva-Trifonova, [4]; застосування нейронних мереж – в роботах Y. Chen та ін. [10], S. Wang та ін. [11], X.-H. Le та ін. [12]. Цей напрям зазнає постійного розвитку, а вчені присвячують значну увагу вдосконаленню та новаторському застосуванню методів та технік. Проте, незважаючи на значні досягнення, в галузі обробки природної мови все ще існують виклики та невизначеність в загальній концепції обробки природної мови.

Формулювання цілей статті: проаналізувати моделі провести комплексний аналіз сучасних моделей глибокого та машинного навчання для оптимізованої обробки природної мови, їх переваги та обмеження, і висвітлити потенційні перспективи для майбутніх досліджень у цій області.

Виклад основного матеріалу.

Оскільки робота з текстами в ручному форматі більше не є ефективною в умовах всебічної автоматизації та цифровізації, є потреба у розгляді нових

інструментів. За останні роки дослідження в області машинного навчання та штучного інтелекту здійснили неймовірний прорив, створено велику кількість архітектури нейронних мереж, які заміщають людину в області обробки відео, фото, аудіо і, звичайно, тексту. Для того, що використовувати різноманітні техніки машинного навчання та глибокого навчання, необхідно привести вхідні дані до відповідного та зручного формату для навчання. Серед найпоширеніших методів обробки природної мови в рамках підготовки тексту можна виділити такі:

1. «Bag of words» («мішок слів») – метод обробки природної мови, який використовується для перетворення тексту на вектор, який далі можна використовувати в алгоритмах машинного навчання. Основна ідея методу полягає в розгляді тексту як набору окремих слів, ігноруючи порядок та граматичні правила, зосереджуючись лише на наявності слів у тексті. Для кожного тексту створюється вектор, який вказує, скільки разів слово зустрічається в тексті [1]. Даний метод простий, але досить ефективний для деяких задач обробки тексту, таких як класифікація документів чи виявлення тем. Однак він не враховує семантичні зв'язки між словами та може втратити інформацію про порядок слів у тексті.

2. Частота термінів – інверсна частота документів (TF-IDF, Term Frequency–Inverse Document Frequency) – метод оцінки ваги термінів/слів, який використовується для оцінки важливості слова в контексті корпусу текстів [2]. Цей метод дозволяє виділити ключові слова в конкретних документах, підсилюючи терміни, які характерні для певного документа, але за рідкістю зустрічаються в інших. Точність класифікації та рівень помилок визначаються за показниками [3]:

Частота термінів (TF, Term Frequency) – визначає, наскільки часто термін/слово зустрічається в конкретному документі. Частота термінів обчислюється за допомогою формули:

$$TF(t,d) = \frac{\text{К-ть появ слова } t \text{ в документі } d}{\text{Загальна к-ть слів в документі } d}, \quad (1)$$

Інверсна частота документів (IDF, Inverse Document Frequency) – вимірює, наскільки інформативним є термін на всій вибірці тексту. IDF обчислюється за допомогою формули 2, а саме логарифму відношення загальної кількості документів N до кількості документів, в яких зустрічається термін/слово t, важливо також додавати 1 до кількості документів, аби уникнути ділення на 0:

$$IDF(t,D) = \log\left(\frac{N}{\text{К-ть документів в збірці, що містять термін } t + 1}\right), \quad (2)$$

Частота термінів – Інверсна частота документів (TF-IDF, Term Frequency–Inverse Document Frequency) – значення перемноження TF на IDF для отримання оцінки важливості терміна/слова в конкретному документі в межах всього документу, розраховується за формулою:

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D), \quad (3)$$

3. Doc2Vec (Paragraph Vector) – розширена модель Word2Vec, яка генерує векторні представлення слів в межах тексту, натомість Doc2Vec генерує векторні представлення цілих абзаців або документів, вона має

кілька ключових компонентів, а її робота базується на певних принципах машинного навчання: PV-DM (Distributed Memory) і PV-DBOW (Distributed Bag of Words).

PV-DM – модель, що намагається передбачити наступне слово у контексті документа, використовуючи його навколишні контекстні слова з додаванням ідентифікатора абзацу (рис. 1).

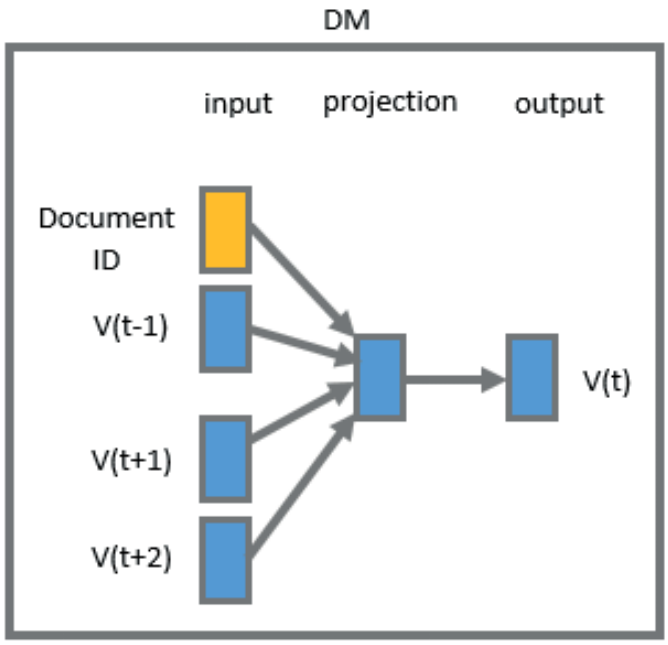


Рис. 1. Інтерпретація PV-DM моделі

Джерело: [4].

PV-DBOW – модель, що намагається передбачити випадково вибрані слова з документа, використовуючи тільки вектор документа, приймає ідентифікатор документа як вхідний (рис. 2).

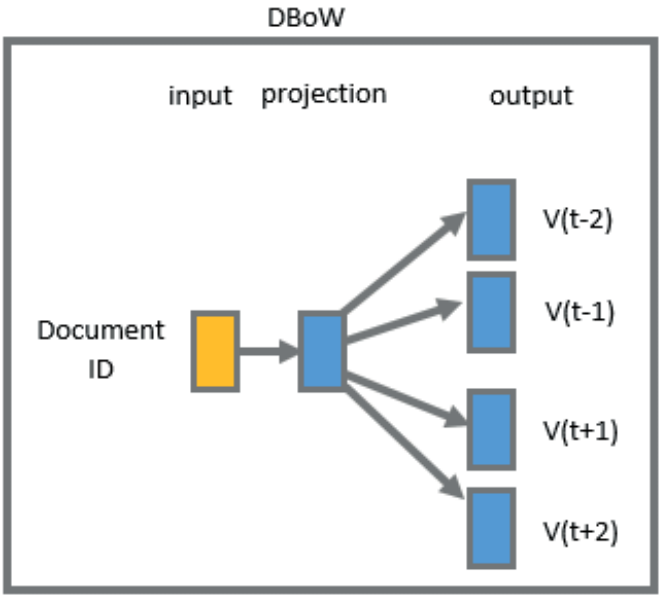


Рис. 2. Інтерпретація PV-DBOW моделі

Джерело: [4].

4. N-gram (N-гами) – використовуються для задач моделювання мови, де метою є передбачення ймовірності слова з огляду на його попередній контекст. У моделі unigram (1-гама) кожне слово розглядається незалежно, тоді як моделі bigram (2-гами) розглядають пари послідовних слів. Моделі Trigram (3-гами) і Higer-Order (N-гам) захоплюють довші залежності. Незважаючи на їх простоту, N-грамові моделі пропонують базову лінію для моделювання мови і корисні в сценаріях, де більш складні моделі є затратними з точки зору реалізації ресурсу [5].

Досить зручною практикою є класифікація текстів, або окремих елементів в тексті. Існує низка методів, якими можна реалізувати дану можливість, оскільки, кожне речення/абзац/документ можливо представити у вигляді вектора чисел та визначати залежності між ними. Алгоритми класифікації, як і будь-які інші алгоритми, можуть відрізнятися своєю

складністю, тобто затребуваними технічними потужностями, які необхідні для виконання всіх ітерацій алгоритму і як результат навчання моделі та її використання, проте на великій навчальній вибірці можуть давати результати з незначною різницею похибок. Нижче представлені два найпопулярніші алгоритми для класифікації текстів/елементів тексту різної складності:

1. Згорткові нейронні мережі (Convolutional Neural Networks, CNN) – ефективно використовуються для класифікації текстів та окремих елементів у тексті. Ці мережі вперше були розроблені для обробки зображень, але їх потужність виявилася корисною і в області обробки природної мови [6]. Використання CNN для класифікації текстів включає такі етапи:

Вхідні Дані:

Текстові дані представляються у вигляді векторів слів або векторних представлень слів (word embeddings). Кожен текст може бути представлений як послідовність векторів, де кожен вектор відповідає слову чи токену в тексті.

Векторизація Тексту:

Зазвичай використовується окремий вектор для кожного слова, якщо використовуються word embeddings. Ці вектори можна об'єднати в матрицю векторів для представлення всього тексту.

Згортка (Convolution):

Згорткові шари використовуються для виявлення локальних залежностей у тексті. Кожен фільтр сканує вхідний текст, шукаючи конкретні патерни слів чи фраз. Згорткові операції дозволяють виділити важливі особливості тексту.

Функція Пулінгу (Pooling):

Після згорткового шару застосовується функція пулінгу, зазвичай максимальний пулінг. Це дозволяє зменшити розмірність отриманих карт ознак і виокремити найбільш важливі особливості.

Повторення Процесу:

Цикл згорткових та пулінгових шарів може повторюватися для виявлення більш абстрактних особливостей у тексті. Це дозволяє нейронній мережі автоматично вивчати різні рівні репрезентації тексту.

Повнозв'язний Шар (Fully Connected Layer):

Після серії згорткових та пулінгових шарів, отримані ознаки передаються до повнозв'язного шару. Цей шар здійснює остаточний процес класифікації, визначаючи до якого класу чи категорії належить текст.

Функція Активації та Функція Втрат:

На останньому шарі використовується функція активації (зазвичай softmax) для отримання ймовірностей належності тексту до різних класів. Функція втрат визначає, наскільки прогноз моделі відрізняється від фактичних міток.

Навчання та Оптимізація:

Мережа навчається за допомогою зворотного поширення помилок (backpropagation) та оптимізаційного алгоритму для адаптації ваг шарів.

Згорткові нейронні мережі для класифікації текстів показують високу ефективність в розпізнаванні локальних та глобальних залежностей у текстах, що робить їх популярними в області обробки природної мови [6].

2. Наївний Баєсівський класифікатор (Naive Bayes) – алгоритм керованого навчання, заснований на застосуванні теореми Баєса з «наївним» припущенням про умовну незалежність між кожною парою ознак, враховуючи значення змінної класу. Теорема Баєса стверджує відповідне

відношення (4), задане змінною класу y і залежним вектором ознак x_1 через x_n.

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}, \tag{4}$$

Використовуючи наївне припущення про умовну незалежність описане виразом (5), для всіх i цей зв'язок спрощується до виразу (6).

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y), \tag{5}$$

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}, \tag{6}$$

Оскільки $P(x_1, \dots, x_n)$ є постійною з урахуванням вхідних даних, можна застосувати правило класифікації:

$$P(y \mid x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i \mid y),$$
$$\Downarrow \tag{7}$$
$$\hat{y} = \arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i \mid y)$$

Використовуючи оцінку апостеріорного максимуму (MAP, Maximum a Posteriori) для оцінки $P(y)$ та $P(x_i|y)$ відносну частоту класу y в навчальному наборі можемо описати як вираз через:

$$\theta_{\text{MAP}} = \arg\max_{\theta} P\,(\theta|D) = \arg\max_{\theta} \frac{P(D|\theta) \cdot P(\theta)}{P(D)}, \qquad (8)$$

Наївні класифікатори Баєса можуть бути надзвичайно швидкими порівняно з більш складними методами. Відокремлення розподілу умовних ознак класу означає, що кожен розподіл може бути незалежно оцінений як одновимірний розподіл. Це допомагає обійти часту проблему з великою розмірністю та полегшити навантаження на виробничі технічні потужності [7].

На сьогодні існують нейронні мережі, які можуть не тільки класифікувати елементи в тексті та залежності між словами, але й генерувати тести та передбачувати наступні елементи речень, навчаючись на послідовностях слів у реченнях. Найпопулярнішими алгоритмами реалізації таких процесів є:

1. Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) - використовують однакові ваги для кожного елемента послідовності, зменшуючи кількість параметрів і дозволяючи моделі узагальнювати послідовності різної довжини. RNN узагальнюються до структурованих даних, крім послідовних, таких як географічні або графічні дані, завдяки своєму дизайну. У стандартних нейронних мережах всі вхідні та вихідні дані незалежні одне від одного. Однак у деяких випадках, наприклад, при прогнозуванні наступного слова у реченні, необхідно враховувати попередні слова, і тому потрібно, щоб попередні слова також запам’ятовувалися. В результаті була створена рекурентна нейронна мережа (RNN), яка

використовує прихований шар для подолання цієї проблеми. Найважливішою частиною RNN є прихований стан, який запам'ятовує конкретну інформацію про послідовність [8].

Рекурентні нейронні мережі, зазвичай, мають архітектуру, при якій на кожен часовий крок t , активація $a^{<t>}$ і результат $y^{<t>}$ виражаються формулою:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{і} \quad y^{<t>} = g_2\left(W_{ya}a^{<t>} + b_y\right), \quad (9)$$

де W_{ax} , W_{aa} , W_{ya} , b_a , b_y *– це коефіцієнти, які виділені тимчасово, і* g_1 , g_2 *– функції активації.*

Графічне представлення архітектури мережі зображено на рисунку 3.

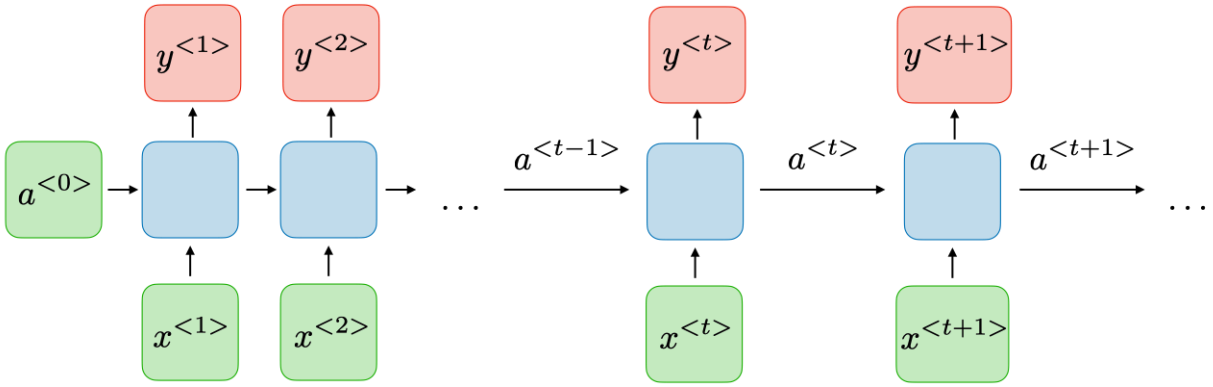


Рис. 3. Архітектура рекурентної нейронної мережі

Джерело: [8].

Найбільш популярними для даної мережі функції активації такі:

- Сигмоїда (Sigmoid): $g(z) = \frac{1}{1+e^{-z}}$
- Гіперболічний тангенс (Tanh): $g(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
- Випрямлений лінійний вузол (RELU): $g(z) = \max(0, z)$

Графічне зображення функцій активації зображено на рисунку 4.

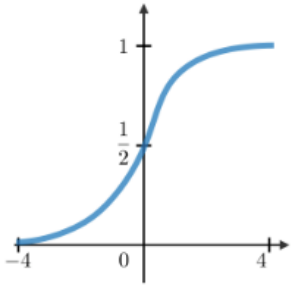
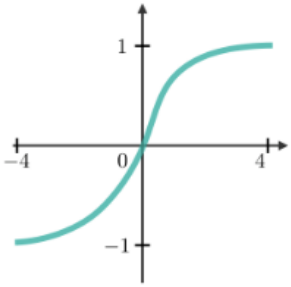
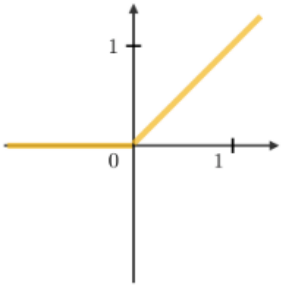
Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

Рис. 4. Функції активації нейронів

Джерело: [9].

Функція активації в нейронних мережах визначає вихідний сигнал (вихідне значення) нейрона в залежності від зважених сум входів та може допомагати введенню нелінійностей у модель.

В залежності від архітектурної структури, рекурентні нейронні мережі можуть бути таких типів:

- Один до Одного (One to One)
- Один до Багатьох (One to Many)
- Багато до Одного (Many to One)
- Багато до Багатьох (Many to Many)

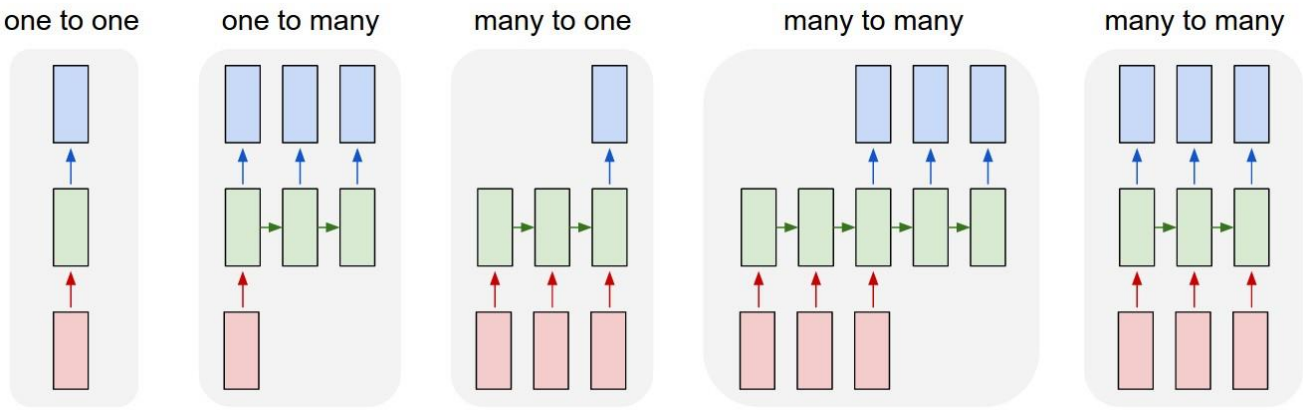


Рис. 5. Типи рекурентних нейронних мереж

Джерело: [8, 9].

1) Один до Одного (One to One) ($T_x = T_y = 1$) – це найбільш базовий та традиційний тип архітектурної структури нейронної мережі, що дає один вихід для одного входу, використовується для вирішення звичайних задач машинного навчання.

2) Один до Багатьох (One to Many) ($T_x = 1,\ T_y > 1$) – це вид архітектури RNN, яка застосовується в ситуаціях, де для одного входу отримується кілька виходів. Основний приклад застосування – генерація музики. В моделях генерації музики RNN використовуються для створення музичного твору (кілька виходів) з однієї музичної ноти (один вхід).

3) Багато до Одного (Many to One) ($T_x > 1,\ T_y = 1$) – зазвичай застосовується в моделях аналізу настрою, цей тип моделі використовується, коли для отримання одного виходу потрібні кілька входів. Наприклад, у моделі текстовий ввід (слова як декілька входів) вказує його фіксований настрій (один вихід). Іншим прикладом може бути модель оцінки фільмів, яка використовує текстові відгуки як вхід, щоб надати рейтинг фільму від 1 до 5.

4) Багато до Багатьох (Many to Many) ($T_x > 1,\ T_y > 1$) – приймає декілька входів і надає декілька виходів. Проте багато-до-багатьох моделі можуть бути двох видів:

4.1. $T_x = T_y$ – це випадок, коли розміри вхідного та вихідного шарів співпадають. Ця форма RNN використовується в розпізнаванні іменованих сутностей.

4.2. $T_x \neq T_y$ – може бути представлена в моделях, де розміри вхідного та вихідного шарів відрізняються. Найбільш поширене застосування такої архітектури RNN багато-до-багатьох зустрічається в машинному перекладі. Наприклад, «I Love you», перекладаються лише двома словами на іспанську – «te amo». Таким чином, моделі машинного перекладу можуть повертати слова більше або менше, ніж вхідний рядок через нерівну архітектуру RNN багато-до-багатьох, яка працює в фоновому режимі [10].

2. Довгострокова короткострокова пам'ять (Long short-term memory, LSTM) – модель, що побудована на базі RNN, однак є доповненою та виключає недолік, який називається «втратою короткострокової пам’яті». Основне обмеження RNN полягає в тому, що дані моделі не можуть запам'ятати дуже довгі послідовності і потрапити в проблему «поступового зникаючого градієнта» [11]. Градієнти функції втрат в нейронних мережах наближаються до нуля, коли додається більше шарів з певними функціями активації, що ускладнює тренування мережі. Мережі LSTM приходять на допомогу для вирішення проблеми «зниклого градієнту». Вони роблять це, ігноруючи/забуваючи непотрібні дані/інформацію в мережі: LSTM буде забувати дані, якщо від інших вхідних даних (слів у попередньому реченні) не надходить корисної інформації. Коли приходить нова інформація, мережа

визначає, яку інформацію проігнорувати, а яку залишити в пам'яті [12]. Розглядати модель зручно, порівнюючи її з RNN (рис. 6):

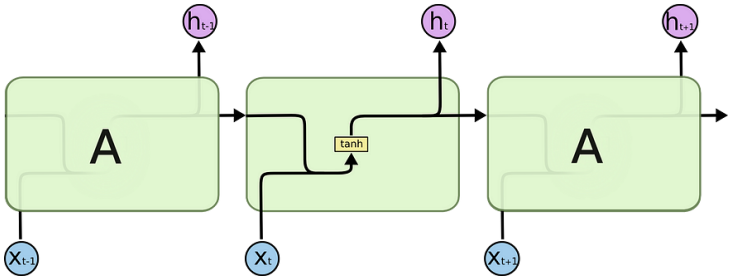


Рис. 6. Графічна інтерпретація логіки моделі RNN з гіперболічним тангенсом, як функція активації

Джерело: [13].

У мережах LSTM, замість простої мережі з однією функцією активації, ми маємо декілька компонентів, які надають мережі можливість забувати та запам'ятовувати інформацію (рис. 7):

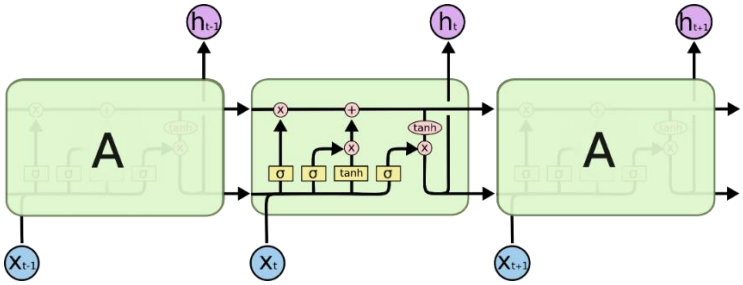


Рис. 7. Графічна інтерпретація логіки моделі LSTM з гіперболічним тангенсом, як функція активації

Джерело: [13].

LSTM мають різні компоненти, розглянемо їх докладніше.

Стан пам’яті (Memory cell) – відповідає за запам'ятовування та забування. На основі контексту введення, тобто що деяку попередню інформацію слід запам'ятовувати, а деяку забувати, і частину нової інформації слід додати до пам'яті. Перша операція (X) - це поелементна операція, яка є множенням стану пам’яті на масив $[-1, 0, 1]$. Інформація,

помножена на 0, буде забута LSTM. Інша операція (+) відповідає за додавання нової інформації до стану пам’яті (рис. 8):

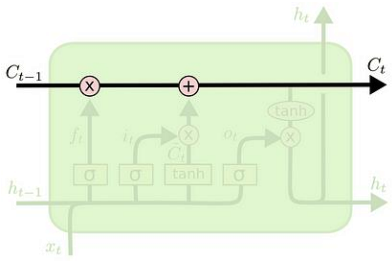


Рис. 8. LSTM Стан пам’яті (Memory cell)

Джерело: [13].

«Забування» (Forget gate) – вирішує, яку інформацію слід «забути», видалити як корегування параметрів моделі, для прийняття цього рішення використовується шар з сигмоїдою, як функцією активації (10), цей шар сигмоїди називається «відсіюванням забуття» (forget gate layer). Він виконує скалярний добуток $h_{(t-1)}$ і $x_{(t)}$ та за допомогою шару сигмоїди виводить число від 0 до 1 для кожного числа в стані пам’яті $C_{(t-1)}$. Якщо вивід «1» – ми зберігаємо цю інформацію, якщо «0» – гарантоване повне забуття інформації. Графічне представлення зображено на рисунку 9.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{10}$$

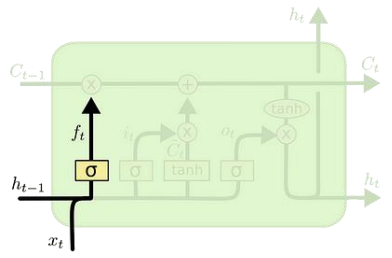


Рис. 9. LSTM забування (Forget Gate)

Джерело: [13].

Вхід (Input gate)

Надає нову інформацію LSTM і вирішує, чи слід цю нову інформацію зберігати в стані пам’яті. Компонент складається з 3-х частин:

1.1. Шар сигмоїди визначає значення для оновлення. Цей шар називається «шар входу».

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{11}$$

1.2. Шар гіперболічного тангенса, як функції активації, створює вектор нових кандидатських значень $\widetilde{C_t}$, які можна додати до стану.

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{12}$$

Графічна інтерпретація кроків 1.1. та 1.2. зображена на рисунку 10.

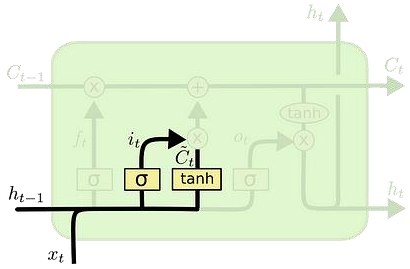


Рис. 10. LSTM Forget Gate

Джерело: [13]

1.3. В результаті ми комбінуємо два виводи як скалярний добуток $i_t \times C_t$ і оновлюємо новий стан пам’яті C_t , який отримується додаванням виводів від «відсіювання забуття» та «входу» (рис. 11).

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}, \tag{13}$$

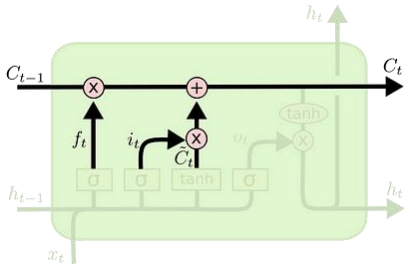


Рис. 11. LSTM новий стан пам’яті (Memory cell)

Джерело: [13].

Вихід (Output gate)

Вихід значення LSTM залежить від нового стану пам’яті. Першим кроком шар функції активації сигмоїди вирішує, які частини стану пам’яті ми будемо виводити,

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{14}$$

другим кроком шар з функцією активації з гіперболічним тангенсом використовується на стані пам’яті для «стискання» значень від -1 до 1, які нарешті множаться на вивід сигмоїдного шару.

$$h_t = o_t * \tanh(C_t), \tag{15}$$

Графічна репрезентація output gate зображена на рисунку 12.

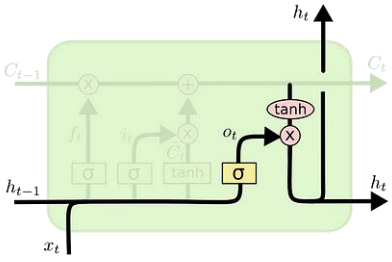


Рис. 12. LSTM Output gate

Джерело: [13].

Переваги LSTM полягають у їх здатності враховувати довгострокові залежності та уникати проблеми вигасання градієнтів, що робить їх ефективними в задачах NLP.

Однак є недоліки. LSTM може бути витратним з точки зору обчислень, особливо при роботі з великими обсягами даних. На дуже довгих послідовностях може виникати проблема пошкодження інформації. Крім того, LSTM має багато параметрів, що може зробити їх вимогливими до даних та потужності обчислень.

Останнім часом з'явилися нові архітектури, такі як трансформери, які у багатьох випадках виявляються більш ефективними в порівнянні з LSTM. Трансформери показують вражаючі результати в NLP завданнях та можуть бути конкурентоспроможними альтернативами у відповідних умовах.

Описані алгоритми представляють лише деякі засоби для класифікації текстів, і вибір конкретного методу може залежати від характеру завдання, обсягу даних та ресурсів. Розвиток глибоких нейронних мереж, наприклад, таких як BERT та інших, відкриває нові можливості для точної та контекстної класифікації текстових даних.

Висновок

У даній статті ми провели комплексний аналіз та порівняння моделей глибокого та машинного навчання в контексті їх застосування для роботи з природною мовою (Natural Language Processing, NLP).

На початку аналізу були розглянуті традиційні методи обробки природної мови, такі як Bag-of-Words та TF-IDF. Ці методи, хоч і надійні, але мають обмежені можливості у розумінні семантичних зв'язків та врахуванні контексту в тексті. Згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN) представляють крок уперед у вирішенні цих проблем,

дозволяючи моделям автоматично вивчати локальні та послідовні залежності у тексті.

Однак переваги глибоких та рекурентних мереж суттєво розширюються з введенням трансформерних архітектур, зокрема BERT та GPT. Ці моделі вражають своєю здатністю враховувати контекст в обидва напрямки та вивчати багаторівневі семантичні зв'язки у тексті. BERT, зокрема, виявляється вельми ефективним у завданнях класифікації текстів та вирішенні завдань NLP.

Проте, на високий рівень точності та ефективності моделей впливають не лише їхні архітектури, але й якість та обсяг навчальних даних. Брак даних може призвести до перенавчання або недонавчання моделі, що негативно вплине на її здатність до адекватного узагальнення.

Крім того, важливими аспектами є швидкість навчання та інференції моделей, їхні витрати на обчислення та можливості масштабування. Враховуючи ці фактори, важливо вибирати архітектури та методи, які відповідають специфічним вимогам та обмеженням проекту.

У майбутньому, напрями досліджень в галузі NLP мають бути спрямовані на подальше вдосконалення архітектур та методів, а також на розвиток нових підходів, спрямованих на вирішення конкретних завдань. Також, увага має зосереджуватися на розробці більш ефективних та економічних методів навчання та використання моделей NLP.

Проте, незважаючи на значні досягнення, в галузі обробки природної мови все ще існують виклики. Наприклад, важливість інтерпретованості моделей, які забезпечують визначення причин та наслідків прийнятих рішень, стає дедалі важливішою. Деякі моделі можуть виявлятися чутливими до контекстних змін, що ускладнює їх використання в змінних умовах.

У кінцевому підсумку, розвиток глибоких та машинних навчальних моделей для роботи з природною мовою продовжує визначати напрямок досліджень у цій області. Швидкий темп інновацій та надзвичайна результативність трансформерних архітектур створюють захоплюючий ландшафт можливостей та викликів. Майбутнє обіцяє ще більше проривів у досягненні високого рівня розуміння та обробки природної мови, що допоможе вирішувати складні завдання та покращувати взаємодію людини з комп'ютерами.

Література

1. Wisam, Q., Musa, M. A., & Bilal, A. (2019), "An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges", *International Engineering Conference,* pp. 200-204. https://doi.org/10.1109/IEC47844.2019.8950616.

2. Sammut, C., Webb, G.I. (2011), "TF–IDF". In C. Sammut, G.I. Webb (Eds.) *Encyclopedia of Machine Learning*. Springer. https://doi.org/10.1007/978-0-387-30164-8_832

3. Jalilifard, A., Caridá, V.F., Mansano, A.F., Cristo, R.S., & da Fonseca, F.P.C. (2021), "Semantic Sensitive TF-IDF to Determine Word Relevance in Documents". In: S.M., Thampi, E., Gelenbe, M., Atiquzzaman, V., Chaudhary, KC. Li, (Eds.) *Advances in Computing and Network Communications. Lecture Notes in Electrical Engineering*, vol 736, Springer. https://doi.org/10.1007/978-981-33-6987-0_27

4. Le, Q.V., & Mikolov, T. (2014), "Distributed Representations of Sentences and Documents", In *Proceedings of the 31st International Conference on Machine Learning*, 32(2), 1188-119. https://doi.org/10.48550/arXiv.1405.4053

5. Georgieva-Trifonova, T. & Duraku, M. (2021), “Research on N-grams feature selection methods for text classification”, *Conference Series: Materials Science and Engineering*, 1031, Article012048. https://doi.org/10.1088/1757-899X/1031/1/012048

6. Vajjala, S., Majumder B., Gupta, A., Surana, H. (2020), *Practical Natural Language Processing: A Comprehensive Guide to Building Real-world NLP Systems*, O'Reilly Media.

7. Phuc, D., & Phung, N. T. K. (2007), “Using Naïve Bayes Model and Natural Language Processing for Classifying Messages on Online Forum,” *IEEE International Conference on Research, Innovation and Vision for the Future*, pp. 247-252. https://doi.org/10.1109/RIVF.2007.369164.

8. Anggraeni, M., Syafrullah, M., & Damanik, H. A. (2019), “Literation Hearing Impairment (I-Chat Bot): Natural Language Processing (NLP) and Naïve Bayes Method”, *Journal of Physics: Conference Series,* 1201, Article012057. https://doi.org/10.1088/1742-6596/1201/1/012057

9. Kmytiuk, T., & Majore, G. (2021), “Time series forecasting of agricultural product prices using Elman and Jordan recurrent neural networks”, *Neuro-Fuzzy Modeling Techniques in Economics*, 10, 67-85. http://doi.org/10.33111/nfmte.2021.067

10. Chen, Y., Gilroy, S., Maletti, A., May, J. & Knight, K. (2018), “Recurrent Neural Networks as Weighted Language Recognizers”. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 2261–2271. https://doi.org/10.48550/arXiv.1711.05408

11. Wang, S. & Jiang, J. (2016), “Learning Natural Language Inference with LSTM”. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1442–1451, https://doi.org/10.48550/arXiv.1512.08849

12. Le, X.-H., Ho, H.V., Lee, G., Jung, S. (2019), “Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting”, *Water*, *11*, 1387. https://doi.org/10.3390/w11071387

13. Saxen, A., (2023), “Introduction to Long Short-Term Memory (LSTM)”, *Analytics Vidhya*

References

1. Wisam, Q., Musa, M. A., & Bilal, A. (2019), “An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges”, *International Engineering Conference*, pp. 200-204. https://doi.org/10.1109/IEC47844.2019.8950616.

2. Sammut, C. and Webb, G.I. (2011), “TF–IDF”, *Encyclopedia of Machine Learning*. Springer. https://doi.org/10.1007/978-0-387-30164-8_832

3. Jalilifard, A., Caridá, V.F., Mansano, A.F., Cristo, R.S., & da Fonseca, F.P.C. (2021), “Semantic Sensitive TF-IDF to Determine Word Relevance in Documents”, *Advances in Computing and Network Communications. Lecture Notes in Electrical Engineering*, vol 736, Springer. https://doi.org/10.1007/978-981-33-6987-0_27

4. Le, Q.V., & Mikolov, T. (2014), “Distributed Representations of Sentences and Documents”, *Proceedings of the 31st International Conference on Machine Learning*, vol. 32(2), pp. 1188-119. https://doi.org/10.48550/arXiv.1405.4053

5. Georgieva-Trifonova, T. & Duraku, M. (2021), “Research on N-grams feature selection methods for text classification”, *Conference Series: Materials Science and Engineering*, vol. 1031, Article012048. https://doi.org/10.1088/1757-899X/1031/1/012048

6. Vajjala, S., Majumder B., Gupta, A. and Surana, H. (2020), *Practical Natural Language Processing: A Comprehensive Guide to Building Real-world NLP Systems*, O'Reilly Media.

7. Phuc, D., & Phung, N. T. K. (2007), “Using Naïve Bayes Model and Natural Language Processing for Classifying Messages on Online Forum”, *IEEE International Conference on Research, Innovation and Vision for the Future*, pp. 247-252. https://doi.org/10.1109/RIVF.2007.369164.

8. Anggraeni, M., Syafrullah, M., & Damanik, H. A. (2019), “Literation Hearing Impairment (I-Chat Bot): Natural Language Processing (NLP) and Naïve Bayes Method”, *Journal of Physics: Conference Series*, vol. 1201, Article012057. https://doi.org/10.1088/1742-6596/1201/1/012057

9. Kmytiuk, T., & Majore, G. (2021), “Time series forecasting of agricultural product prices using Elman and Jordan recurrent neural networks”, *Neuro-Fuzzy Modeling Techniques in Economics*, vol. 10, pp. 67-85. http://doi.org/10.33111/nfmte.2021.067

10. Chen, Y., Gilroy, S., Maletti, A., May, J. & Knight, K. (2018), “Recurrent Neural Networks as Weighted Language Recognizers”, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 2261–2271. https://doi.org/10.48550/arXiv.1711.05408

11. Wang, S. & Jiang, J. (2016), “Learning Natural Language Inference with LSTM”, In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1442-1451, https://doi.org/10.48550/arXiv.1512.08849

12. Le, X.-H., Ho, H.V., Lee, G. and Jung, S. (2019), “Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting”, *Water*, vol. 11, 1387. https://doi.org/10.3390/w11071387

13. Saxen, A., (2023), “Introduction to Long Short-Term Memory (LSTM)”, *Analytics Vidhya*

Стаття надійшла до редакції 09.02.2024 р.