

ТЕХНІЧНІ НАУКИ

DOI: <https://doi.org/10.33216/1998-7927-2023-278-2-96-105>

УДК 004.9

МОДЕЛІ МАШИННОГО НАВЧАННЯ
ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ

Шумова Л.О., Рязанцев О.І., Покришка С.А.

MACHINE LEARNING MODELS
FOR THE FORMATION OF RECOMMENDATIONS

Shumova L., Ryazantsev O., Pokrishka S.

У статті представлено експериментальне дослідження моделей глибоких нейронних мереж для формування релевантних рекомендацій користувачу інтернет-ресурсів. Рекомендаційна система представляє програмний засіб, що використовує певний алгоритм фільтрації та наявну інформацію про потреби користувача, щоб рекомендувати йому релевантний набір об'єктів, які будуть найбільш корисними для нього.

Аналіз останніх досліджень і публікацій в області впровадження рекомендаційних систем показав, що підвищення якості пропозицій рекомендаційних систем на основі методів машинного навчання є актуальною задачею. Використання нейронних мереж у рекомендаційних системах може підвищити ефективність та зручність використання цих систем. Метою дослідження є підвищення якості пропозицій рекомендаційних систем на основі методів машинного навчання.

В ході досліджень систематизовано сукупність етапів та визначено методологію побудови ефективної рекомендаційної системи з використанням методів машинного навчання. Визначені засоби, формалізовані етапи, представлена технічна блок-схема розробки нейронної рекомендаційної системи. Побудовано моделі двох систем з використанням колаборативної фільтрації й глибокої матричної факторизації. Проведено аналіз продуктивності цих систем за допомогою таких показників, як точність, повнота та нормалізований дисконтований кумулятивний виграш.

Дослідження проведені з використанням таких алгоритмів оптимізації: SGD, RMSprop, ADAdelta і FTRL. Результати експериментального дослідження моделей глибоких нейронних мереж для формування рекомендацій у різних сценаріях показали, що

продуктивність нейронних систем рекомендацій може сильно відрізнятися в залежності від типу використовуваної моделі пошуку, кількості та якості даних, а також архітектури та методу навчання мережі.

За результатами проведених експериментів визначено оптимальні алгоритми навчання моделі нейронмережевої рекомендаційної системи для вирішення певної задачі в залежності від характеру вихідних даних.

Експериментальне дослідження проведено за допомогою мови програмування Python, з використанням бібліотеки TensorFlow.

У роботі використано вільно доступні набори даних про рейтинги фільмів наданих користувачами сайту MovieLens.

Ключові слова: рекомендаційна система, методи фільтрації, алгоритми машинного навчання, оцінка продуктивності.

Вступ. Рекомендаційні системи використовуються на веб-сайтах та у програмних додатках для створення списків рекомендацій користувачам на основі їх попередніх дій, наприклад, переглянутого контенту, виставлених оцінок, тощо. Працюючи з певним типом інформації та системою фільтрів, рекомендаційні системи прогнозують вподобання користувачів для впровадження персоналізованих рекомендацій [1].

Постановка проблеми. Алгоритми роботи рекомендаційних систем можуть використовувати різні методи фільтрації даних, зокрема,

контентну, колаборативну, тощо. Кожен з видів фільтрації має багато різних способів реалізації та модифікацій, але всі вони мають свої переваги і недоліки. [2, 3].

Таким чином, рекомендаційна система представляється у вигляді програмного засобу, що використовує певний алгоритм фільтрації та наявну інформацію про потреби користувача, щоб рекомендувати йому релевантний набір об'єктів, які будуть найбільш корисними для нього. При цьому важливим алгоритмічним завданням є пошук чіткої логіки того, що сподобається користувачу, тобто треба знайти шаблони в даних і застосувати ці знання для передбачення нових даних. Потужним механізмом здатним вивчати та адаптуватися до різних типів даних і вирішувати складні завдання є штучні нейронні мережі.

Тому для прогнозування уподобань користувачів і формування персоніфікованих рекомендацій доцільно використовувати нейронні рекомендаційні мережі, які можуть знаходити приховані зв'язки між поведінкою користувача й характеристиками об'єкта та видавати релевантні пропозиції.

Аналіз останніх досліджень і публікацій.

Проведений аналіз досліджень показав, що до актуальних задач впровадження рекомендаційних систем є підвищення якості рекомендацій - створення релевантних рекомендацій для даного користувача. В останній час свою ефективність у різних рекомендаційних задачах показали нейронні мережі. [4-6].

Однією з основних переваг використання нейронних мереж у рекомендаційних системах є їхня здатність обробляти великі обсяги даних та працювати з високорозмірними вхідними просторами. Це дозволяє їм уловлювати тонкі закономірності та взаємозв'язки, які можуть бути неочевидними при використанні більш простих моделей. Крім того, нейронні мережі здатні адаптуватися до змін уподобань користувачів та характеристик об'єктів з часом, що підвищує їхню придатність для динамічних середовищ [7].

Використання технологій нейромереж має велику популярність через доволі точні розрахунки та гнучкість у налаштуванні. Загалом використання нейронних мереж у рекомендаційних системах може підвищити ефективність та зручність використання цих систем [8].

Багато великих компаній, таких як AirBnB, Facebook, Google, Home Depot, LinkedIn та Pinterest, діляться своїм досвідом використання глибокого навчання для подальшого підвищен-

ня якості рекомендацій. Ковінгтон та ін. представили глибокий алгоритм рекомендацій на основі нейронної мережі для відеорекомендацій на YouTube. Cheng та співавт запропонували систему рекомендацій додатків для Google Play із широкою та глибокою моделлю. Усі ці моделі витримали онлайн-тестування та продемонстрували суттєве покращення порівняно з традиційними моделями.

Незважаючи на те, що створені та активно використовуються багато методів та алгоритмів формування рекомендацій на основі глибокого навчання, не завжди зрозуміло, який краще буде працювати на конкретному веб-ресурсі. Тому дослідження алгоритмів глибокого навчання у розробці рекомендаційних систем є актуальними.

Мета статті. Метою дослідження є підвищення якості пропозицій рекомендаційних систем на основі методів машинного навчання.

Завдання дослідження визначено наступним чином:

- формалізація етапів та визначення методології побудови ефективної рекомендаційної системи з використанням методів машинного навчання: визначення засобів, формалізація етапів, розробка технічної блок-схеми процесу;
- експериментальне дослідження моделей глибоких нейронних мереж для формування рекомендацій у різних сценаріях; порівняння точності, швидкості обчислення для двох нейромережевих моделей рекомендаційної системи: моделі нейронної колаборативної фільтрації та моделі глибокої матричної факторизації.

Основна частина. Задачу розробки нейронної рекомендаційної системи запропоновано розбити на декілька етапів: збір і систематизація даних, попередня обробка даних, вибір моделі нейронної мережі, навчання моделі, уточнення моделі, розгортання моделі. Блок-схема процесу представлена на рис. 1.

По-перше, необхідно зібрати дані про користувачів, об'єкти, що їх цікавлять, та їх взаємодію (наприклад, рейтинги чи відгуки). Також потрібно організувати ці дані у форматі, який підходить для навчання нейронної мережі, наприклад, у вигляді матриці або списку словників. Дані для нейронної рекомендаційної системи можуть надходити з різних джерел, таких як:

- оцінки чи відгуки користувачів про об'єкти, що їх цікавлять. Ці дані можуть бути зібрані за допомогою опитувань або відстеження активності користувачів (наприклад, кліки, перегляди, покупки);

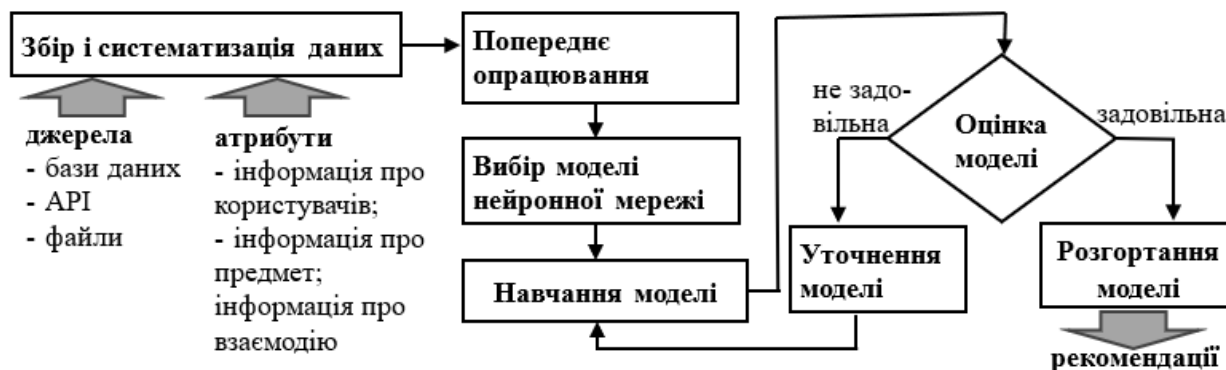


Рис. 1. Блок-схема розробки нейронної рекомендаційної системи

- метадані про об'єкти. Ці дані включають характеристики об'єктів, що рекомендуються, такі як їх назви, описи, категорії тощо;

- метадані користувача. Ці дані включають характеристики користувачів, яким рекомендуються об'єкти, такі як їх вік, стать, місцезнаходження тощо.

Ці дані можуть бути зібрані та збережені у базі даних або у файлі (наприклад, CSV-файлі).

Варто також зазначити, що дані, які використовуються для навчання рекомендаційної системи, мають бути репрезентативними стосовно даних, з якими система зіткнеться в процесі роботи.

У цій роботі використано вільно доступні набори даних про рейтинги фільмів наданих користувачами сайту MovieLens [9], які збирала дослідницька група GroupLens. Це популярний набір даних для тестування алгоритмів спільної фільтрації.

Перш ніж використовувати дані для навчання рекомендаційної системи може знадобитися попередня обробка даних шляхом виконання таких завдань, як нормалізація значень, вирівнювання відсутніх даних або видалення винятків. Для підготовки даних потрібно:

- попередньо обробити та очистити дані в міру необхідності. Це може включати такі завдання, як видалення пропущених значень, перетворення даних у відповідний формат та нормалізація числових даних;

- розділити дані на навчальний та тестовий набори;

- перетворити дані у формат, який може бути використаний нейронною мережею. Це може включати створення одноточкових кодувань для категоріальних змінних та масштабування числових змінних до певного діапазону.

При побудові нейронної мережі, потрібно вибрати архітектуру моделі, яка підходить для даних та конкретного завдання. Існує безліч різних типів моделей, які можуть бути викорис-

тані у рекомендаційних системах, включаючи моделі колаборативної фільтрації [10], моделі фільтрації за вмістом [11] та гібридні моделі [12]. Фільтрація за вмістом та колаборативна фільтрація - два підходи, що широко використовуються в нейронних рекомендаційних системах, а нейроматрична факторизація - більш сучасна розробка, яка поєднує в собі сильні сторони обох методів, що робить її ефективним методом рекомендацій для великомасштабних, розріджених наборів даних [13].

При виборі моделі необхідно враховувати такі фактори, як обсяг та якість даних, складність завдання та доступні обчислювальні ресурси. Щоб знайти кращу модель, доцільно випробувати кілька моделей і порівняти їхню продуктивність [14, 15].

У представленій роботі досліджені дві моделі: модель колаборативної фільтрації й модель глибокої матричної факторизації.

Навчання моделі включає поділ даних на навчальний та перевірочний набори, а також використання навчального набору даних для налаштування ваг (зв'язків між нейронами) та зсувів (порогів активації нейронів).

Є кілька способів навчання моделі нейронної мережі, зокрема:

- стохастичний градієнтний спуск (SGD). Це простий і ефективний метод, який передбачає ітеративне оновлення ваг та зсувів моделі в напрямку, що мінімізує функцію втрат [16];

- міні-пакетний градієнтний спуск. Цей метод схожий на SGD, але він оновлює ваги та зміщення моделі на основі невеликої "міні-партії" даних, а не всього набору даних. Це може допомогти моделі сходиться швидше і може бути стабільнішою, ніж SGD;

- пакетна нормалізація. Ця техніка включає нормалізацію активацій моделі на кожному шапі для зменшення внутрішнього зсуву коваріації і прискорення навчання [17].

Для навчання моделі також необхідно вибрати відповідну функцію втрат і оптимізатор. Функція втрат вимірює різницю між прогнозованим виходом та справжнім виходом, а оптимізатор налаштовує вагові коефіцієнти та зсуви моделі, щоб мінімізувати втрати. Існує безліч різних функцій втрат та оптимізаторів, і вибір залежить від конкретних характеристик даних та завдання.

Дослідження проведені з використанням таких алгоритмів оптимізації: SGD [16], RMSprop [18], ADAdelta і FTRL.

Оцінка моделі полягає у використанні даних для перевірки продуктивності моделі. Це може включати обчислення таких показників, як точність (прецизійність) й повнота [19].

У цій роботі для оцінки продуктивності моделі використовується три метрики: NDCG (Normalized discounted cumulative gain), Precision, Recall [20, 21].

Якщо ефективність моделі виявиться незадовільною, то може знадобитися точне налаштування моделі шляхом коригування гіперпараметрів або зміни архітектури нейронної мережі.

Гіперпараметри - це параметри, що управляють поведінкою моделі, такі як швидкість навчання, розмір партії чи кількість прихованих одиниць. Налаштування цих гіперпараметрів може вплинути на здатність моделі до навчання та узагальнення нових даних. Можна використовувати такі методи, як пошук по сітці або випадковий пошук, щоб знайти найкращі значення гіперпараметрів для моделі.

Також може знадобитися змінити архітектуру моделі, якщо вона надає не релевантні рекомендації. Це може включати додавання або видалення шарів, зміну кількості нейронів у кожному шарі або зміну функцій активації. Експериментування з різними архітектурами допоможе знайти оптимальну для конкретного завдання.

Після навчання та тонкого налаштування моделі потрібно розгорнути її у виробничому середовищі, щоб надавати рекомендації користувачам.

При створенні рекомендаційної системи важливо враховувати масштабованість та надійність розгортання. Необхідно переконатися, що модель здатна обробляти великий обсяг за-

питів та відновлюватись після збоїв чи помилок. Також може знадобитися врахувати питання безпеки та конфіденційності, такі як захист даних або запобігання несанкціонованому доступу до моделі.

Надалі необхідно контролювати і підтримувати розгорнуту модель, щоб вона продовжувала адекватно працювати. Це може включати такі завдання, як моніторинг продуктивності моделі, оновлення моделі новими даними або перенавчання моделі у разі зниження її продуктивності.

Використані технології. На даний момент розроблено безліч бібліотек нейронних мереж з відкритим вихідним кодом, серед них TensorFlow від компанії Google; PyTorch - розроблена Facebook; Keras – розроблена спільнотою дослідників, основним автором є Франсуа Шолле (Google); Theano – розроблений групою машинного навчання Монреальського університету; Caffe - розроблений Центром зору та навчання Берклі (Каліфорнійський університет у Берклі).

Ці бібліотеки надають ряд інструментів та можливостей для побудови та навчання нейронних мереж, включаючи підтримку різних мережевих архітектур, алгоритмів оптимізації та апаратних прискорювачів (наприклад, підтримку GPU).

Представлене експериментальне дослідження проведено за допомогою мови програмування Python, з використанням бібліотеки TensorFlow.

Результати експерименту. Проведено серію експериментів з метою визначення оптимальних алгоритмів навчання моделі, що забезпечують максимальну точність та повноту рекомендацій. Для тесту швидкості навчання моделі було взято файл ratings.csv з movieLens з 25 мільйонами оцінок. Попередня обробка не потрібна, оскільки вона вже зроблена. Для створення нейромережі використовується мова програмування Python із бібліотекою Keras (TensorFlow).

Модель нейронної рекомендаційної системи з використанням колаборативної фільтрації.

Модель нейронної рекомендаційної системи з використанням колаборативної фільтрації показана на рисунку 2.

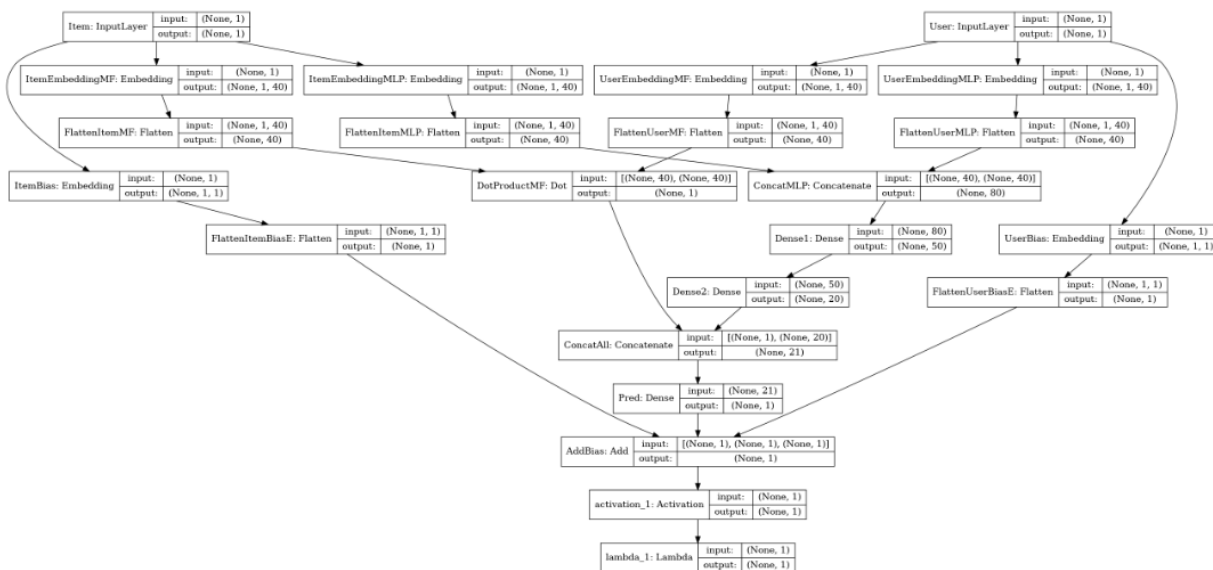


Рис. 2. Модель нейронної рекомендаційної системи з використанням колаборативної фільтрації

Таблиця 1

Оцінка ефективності моделі нейронної колаборативної фільтрації

Метрики	3 епоха	5 епоха	10 епоха	20 епоха	50 епоха
NDCG@k	0.058008	0.065452	0.064215	0.058722	0.052205
Precision@k	0.041357	0.047614	0.046341	0.041463	0.040085
Recall@k	0.018205	0.023217	0.020926	0.020315	0.017497

Графік навчання моделі нейронної колаборативної фільтрації показаний на рисунку 3.

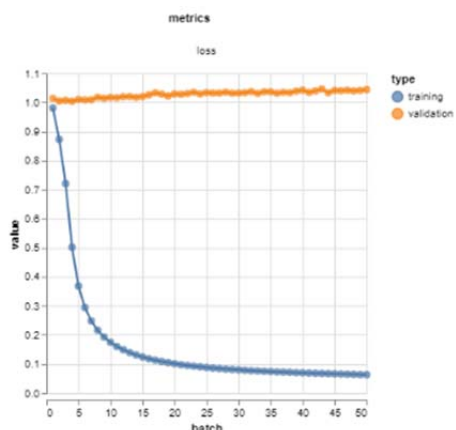


Рис. 3. Графік навчання моделі нейронної колаборативної фільтрації

Оцінка ефективності моделі. Для оцінки ефективності моделі використані класичні метрики: Precision@k (1) - вимірює частку релевантних рекомендованих елементів у списку рекомендацій розміру k. Релевантними вважаємо ті, з якими користувач взаємодіяв за тестовий період; Recall@k (2) - взаємний ранг, визна-

чення збігів ґрунтується на частці відповідних рекомендованих елементів у списку рекомендацій розміру k і загальній кількості відповідних елементів: NDCG@k (3) - нормалізований дисконтований кумулятивний виграш/приріст [20, 21].

$$Precision@k = \frac{\text{кількість відповідних рекомендованих } k \text{ елементів}}{k} \tag{1}$$

$$Recall@k = \frac{\text{кількість відповідних рекомендованих } k \text{ елементів}}{\text{загальна кількість відповідних елементів}} \tag{2}$$

$$NDCG@k = \frac{DCG@k}{IDCG@k}, \tag{3}$$

де IDCG@k - дисконтований сукупний прибуток, що дисконтує «цінність» кожного з відповідних елементів залежно від його рангу; IDCG@k - «ідеальний» алгоритм рекомендацій, тобто алгоритм, в якому кожен відповідний елемент розташовується на початку списку. Оптимальним виявилось навчання приблизно на 5-10 епохах (табл. 1).

Оцінити роботу оптимізаторів моделі на основі нейронної колаборативної фільтрації легко за рисунком 4.

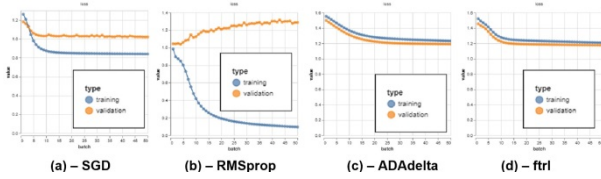


Рис. 4. Оцінка оптимізаторів моделі нейронної колаборативної фільтрації

Метод глибокої факторизації. Глибока факторизація [22] – це метод машинного навчання, який поєднує сильні сторони колаборативної фільтрації та методів матричної факторизації для рекомендаційних систем. Вона передбачає використання глибоких нейронних мереж для вивчення прихованих (латентних) факторів користувачів та об’єктів, що пропонуються, та складання рекомендацій на основі вивчених уявлень. Глибока факторизація показала свою ефективність у підвищенні точності рекомендацій, особливо при роботі з великомасштабними та розрідженими даними. Модель нейронної рекомендаційної системи з використанням глибокої матричної факторизації представлена на рис. 5.

Графік навчання моделі глибокої матричної факторизації показаний на рисунку 6.

Оцінка продуктивності моделі глибокої матричної факторизації за обраними метриками (NDCG, Precision, Recall) представлено у таблиці 2. Оптимальним виявилось навчання приблизно на 3 епосі.

Оцінити роботу оптимізаторів моделі на основі глибокої матричної факторизації можна за рисунком 7.

Колаборативний метод вимагає близько 5 годин 30 хвилин для навчання моделі на такому обсязі файлу, в той же час матричний підхід зажадав близько 7 годин для навчання.

Аналіз ефективності навчання двох моделей показав таке:

- швидкість навчання колаборативної моделі становила загалом 1 хвилина 15 секунд для 50000 елементів, для моделі матричної факторизації - 1 хвилина для 50000 елементів;
- колаборативний підхід на використуваному наборі даних досяг найкращих показників точності (NDCG) у сценарії повторного виклику. Підхід має кращу інтерпретованість, оскільки він заснований на простому точковому добутку вкладень користувача та елемента;

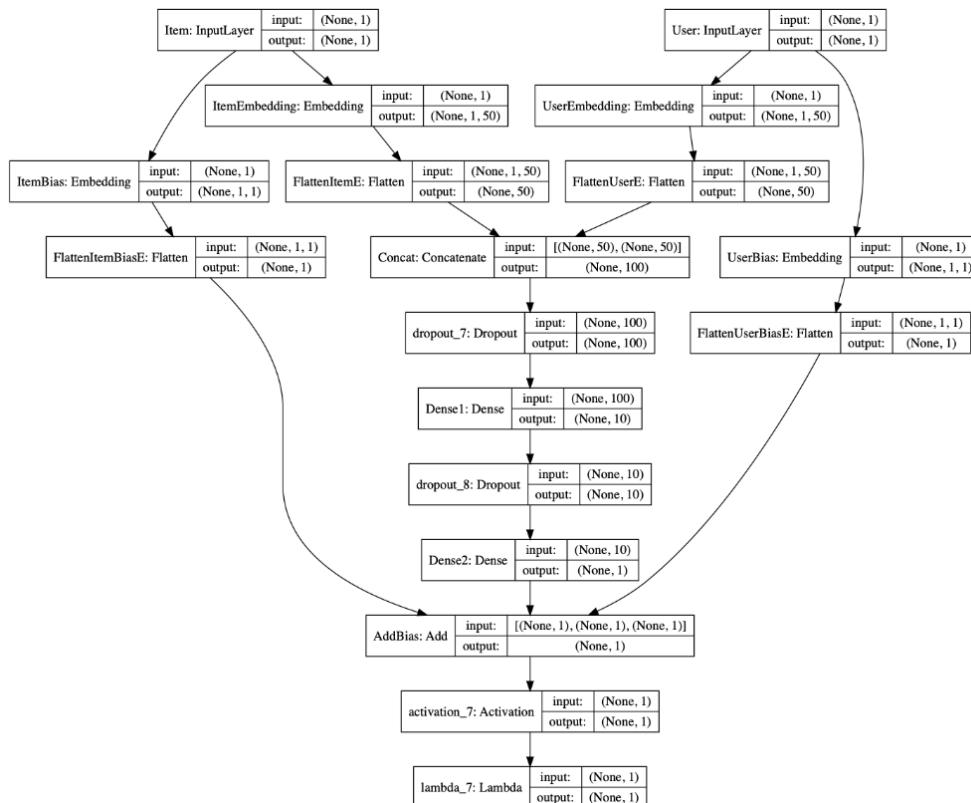


Рис. 5. Модель глибокої матричної факторизації

Таблиця 2

Оцінка моделі глибокої матричної факторизації

Метрики	3 епоха	5 епоха	10 епоха	20 епоха	50 епоха
NDCG@K	0.026713	0.025046	0.016059	0.007580	0.00460
Precision@K	0.018452	0.018558	0.013680	0.006257	0.003499
Recall@K	0.013883	0.013496	0.010131	0.004782	0.00271

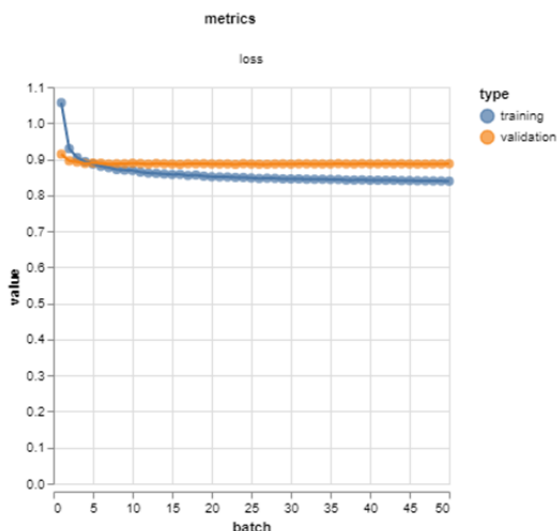


Рис. 6. Графік навчання моделі глибокої матричної факторизації

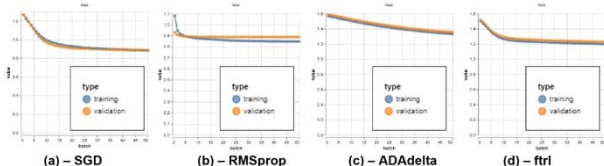


Рис. 7. Оцінка оптимізаторів моделі глибокої матричної факторизації

- підхід з глибокою матричною факторизацією здатний обробляти більш розріджені дані та краще масштабуватись на великі набори даних;

- обчислювальний час та необхідні ресурси були вищими для підходу глибокої матричної факторизації, але це може бути прийнятно, якщо переваги у продуктивності значні.

Результати експериментального дослідження моделей глибоких нейронних мереж для формування рекомендацій у різних сценаріях—показали, що продуктивність нейронних систем рекомендацій може сильно відрізнятись в залежності від типу використовуваної моделі пошуку, кількості та якості даних, а також архітектури та методу навчання мережі.

Висновки. Побудовано моделі двох систем з використанням колаборативної фільтрації й

глибокої матричної факторизації. та проведено аналіз їхньої продуктивності за допомогою таких показників, як точність, повнота та нормалізований дисконтований кумулятивний вигреш (NDCG).

В цілому, колаборативна нейронна рекомендаційна система може працювати краще, ніж система матричної факторизації, з точки зору точності та повноти, в той час як система матричної факторизації може працювати краще з точки зору NDCG.

Колаборативна рекомендаційна нейронна система може краще справлятися з проблемою холодного старту в порівнянні з системою матричної факторизації. Це пов'язано з тим, що колаборативна система враховує історію взаємодії користувача та предмета і схожість між користувачами та предметами, тоді як система матричної факторизації покладається виключно на матричну факторизацію для складання прогнозів.

Система матричної факторизації може бути більш масштабованою порівняно з колаборативною системою, оскільки вона може ефективно обробляти великі набори даних. З іншого боку, колаборативна система може стати повільною та неефективною зі збільшенням кількості користувачів та елементів.

Колаборативна система може працювати краще в розріджених наборах даних у порівнянні з системою матричної факторизації, оскільки вона може обробляти дані, що відсутні, і давати рекомендації на основі наявних даних. З іншого боку, система матричної факторизації може зазнавати труднощів у розріджених наборах даних, що призведе до низької продуктивності.

Вибір нейронної системи рекомендацій залежить від конкретних вимог застосування та характеристик даних, а вибір метрики потрібно робити з урахуванням конкретної предметної області та попередньо обробляти дані.

Ретельно розглянувши ці фактори та оцінивши продуктивність різних методів рекомендацій, можна спроектувати та розробити ефективні нейронні системи рекомендацій для різних додатків.

Література

1. Мелешко С.В. Дослідження методів побудови рекомендаційних систем в мережі Інтернет / С.В. Мелешко, Г.С. Семенов, В.Д. Хох. // Збірник наукових праць "Системи управління, навігації та зв'язку". Випуск 1(47). – Полтава: ПНТУ ім. Ю. Кондратюка. – 2018. – С. 131–136.
2. Recommender systems. Algorithms [Електронний ресурс]. Режим доступу: https://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/algorithms.html (дата звернення: 30.07.2023)
3. Покришка С.А., Шумова Л.О. Удосконалення рекомендаційної веборієнтованої системи з використанням колаборативної фільтрації. // Вісник Національного технічного університету «ХПІ». Збірник наукових праць. Серія: Інформатика та моделювання. – Харків : НТУ «ХПІ». – 2021. – № 1 (5). – С. 115 – 123.
4. Zhang, S., Yao, L., Sun, A., Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. (CSUR)* 52(1), 2019, pp. 1-38.
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017*; pp. 173–182.
6. Vasileios Perifanis, Pavlos S. Efraimidis. *Federated Neural Collaborative Filtering, Knowledge-Based Systems, Volume 242, 2022, 108441, ISSN 950-7051*, <https://doi.org/10.1016/j.knosys.2022.108441>.
7. Tian Y., Peng S., Zhang X., Rodemann T., Tan K., Jin Y. A Recommender System for Metaheuristic Algorithms for Continuous Optimization Based on Deep Recurrent Neural Networks. *IEEE Transactions on Artificial Intelligence*, vol. 1, № 1, pp. 5-18, Aug. 2020, doi: 10.1109/TAI.2020.3022339.
8. Melville P., Sindhvani V. *Recommender Systems*, Encyclopedia of Machine Learning, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010.
9. <https://grouplens.org/datasets/movielens/>
10. Statistical Analysis of K-Nearest Neighbor Collaborative Recommendation URL: <https://arxiv.org/pdf/1010.0499.pdf> (Last accessed: 20.04.2023).
11. Isinkaye F. O. Recommendation systems: Principles, methods and evaluation / F. O. Isinkaye, Y. O. Fola-jimi, 75 B. A. Ojokoh // *Egyptian Informatics Journal*. – Vol. 16. – 2015. – P.261–273.
12. Tegene A., Liu Q., Gan Y., Dai T., Leka H., Aye-new M. Deep Learning and Embedding Based Latent Factor Model for Collaborative Recommender Systems. *Appl. Sci.* 2023, 13, 726. <https://doi.org/10.3390/app13020726>
13. Xiaoyu Du et al. Deep Matrix Factorization Models for Recommender Systems. URL: <https://www.ijcai.org/proceedings/2017/0447.pdf> (Last accessed: 20.04.2023)
14. Yamashita A., Kawamura H., Suzuki K. Similarity Computation Method for Collaborative Filtering Based on Optimization // *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2010. Vol. 14. № 6. P. 654–660.
15. Cimini G., Medo M., Zhou T., Wei D., Zhang Y.-C. Heterogeneity, quality, and reputation in an adaptive recommendation model // *The European Physical Journal B*. 2011. Vol. 80. № 2. P. 201–208.
16. Ruder S. An overview of gradient descent optimization algorithms (2017) <https://arxiv.org/abs/1609.04747> = <https://www.ruder.io/optimizing-gradient-descent/>
17. Ioffe S., Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift (2015) <https://arxiv.org/abs/1502.03167>
18. Hinton G. Overview of mini-batch gradient descent (2012) https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
19. Shani G., Gunawardana A. Evaluating recommender systems, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira, P. Kantor (Springer, Berlin, 2011), pp. 257–297.
20. Murrell T. Evaluating Recommendation Systems - Precision@k, Recall@k, and R-Precision (2023) <https://www.shaped.ai/blog/evaluating-recommendation-systems-part-1>
21. Murrell T. Evaluating recommendation systems - (mAP, MMR, NDCG) <https://www.shaped.ai/blog/evaluating-recommendation-systems-map-mmr-ndcg>
22. He X., et al. Deep Factorization Machines for Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, 2017.

References

1. Meleshko E.V. Doslidzhennya metodiv pobudovy rekomendatsiynykh system v merezhi Internet / E.V. Meleshko, H.S. Semenov, V.D. Khokh. // Zbirnyk naukovykh prats' "Systemy upravlinnya, navihatsiyi ta zv'yazku". Vypusk 1(47). – Poltava: PNTU im. YU. Kondratyuka. – 2018. – S. 131–136.
2. Recommender systems. Algorithms [Elektronnyy resurs]. Rezhym dostupu: https://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/algorithms.html (data zvernennya: 30.07.2023)
3. Pokryshka S.A., Shumova L.O. Udoskonalennya rekomendatsiynoyi veboriyentovanoyi systemy z vykorystanniam kolaboratyvnoyi fil'tratsiyi. //

- Visnyk Natsional'noho tekhnichnoho universytetu «KHPI». Zbirnyk naukovykh prats'. Seriya: Informatyka ta modelyuvannya. – Kharkiv : NTU «KHPI». – 2021. – № 1 (5). – S. 115 – 123.
4. Zhang, S., Yao, L., Sun, A., Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. (CSUR)* 52(1), 2019, pp. 1-38.
 5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017*; pp. 173–182.
 6. Vasileios Perifanis, Pavlos S. Efraimidis. *Federated Neural Collaborative Filtering, Knowledge-Based Systems, Volume 242, 2022, 108441, ISSN 950-7051*, <https://doi.org/10.1016/j.knosys.2022.108441>.
 7. Tian Y., Peng S., Zhang X., Rodemann T., Tan K., Jin Y. A Recommender System for Metaheuristic Algorithms for Continuous Optimization Based on Deep Recurrent Neural Networks. *IEEE Transactions on Artificial Intelligence*, vol. 1, № 1, pp. 5-18, Aug. 2020, doi: 10.1109/TAI.2020.3022339.
 8. Melville P., Sindhvani V. *Recommender Systems, Encyclopedia of Machine Learning*, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010.
 9. <https://grouplens.org/datasets/movielens/>
 10. Statistical Analysis of K-Nearest Neighbor Collaborative Recommendation URL: <https://arxiv.org/pdf/1010.0499.pdf> (Last accessed: 20.04.2023).
 11. Isinkaye F. O. Recommendation systems: Principles, methods and evaluation / F. O. Isinkaye, Y. O. Fola-jimi, 75 B. A. Ojokoh // *Egyptian Informatics Journal*. – Vol. 16. – 2015. – P.261–273.
 12. Tegene A., Liu Q., Gan Y., Dai T., Leka H., Aye-new M. Deep Learning and Embedding Based Latent Factor Model for Collaborative Recommender Systems. *Appl. Sci.* 2023, 13, 726. <https://doi.org/10.3390/app13020726>
 13. Xiaoyu Du et al. Deep Matrix Factorization Models for Recommender Systems. URL: <https://www.ijcai.org/proceedings/2017/0447.pdf> (Last accessed: 20.04.2023)
 14. Yamashita A., Kawamura H., Suzuki K. Similarity Computation Method for Collaborative Filtering Based on Optimization // *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2010. Vol. 14. № 6. P. 654–660.
 15. Cimini G., Medo M., Zhou T., Wei D., Zhang Y.-C. Heterogeneity, quality, and reputation in an adaptive recommendation model // *The European Physical Journal B*. 2011. Vol. 80. № 2. P. 201–208.
 16. Ruder S. An overview of gradient descent optimization algorithms (2017) <https://arxiv.org/abs/1609.04747> = <https://www.ruder.io/optimizing-gradient-descent/>
 17. Ioffe S., Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift (2015) <https://arxiv.org/abs/1502.03167>
 18. Hinton G. Overview of mini-batch gradient descent (2012) https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
 19. Shani G., Gunawardana A. Evaluating recommender systems, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira, P. Kantor (Springer, Berlin, 2011), pp. 257–297.
 20. Murrell T. Evaluating Recommendation Systems - Precision@k, Recall@k, and R-Precision (2023) <https://www.shaped.ai/blog/evaluating-recommendation-systems-part-1>
 21. Murrell T. Evaluating recommendation systems - (mAP, MMR, NDCG) <https://www.shaped.ai/blog/evaluating-recommendation-systems-map-mmr-ndcg>
 22. He X., et al. *Deep Factorization Machines for Recommender Systems*. *IEEE Transactions on Knowledge and Data Engineering*, 2017.

Shumova L., Ryazantsev O., Pokrishka S. Machine learning models for the formation of recommendations

The article presents an experimental study of deep neural network models for generating relevant recommendations for each user of Internet resources. A recommender system is a software tool that uses a specific filtering algorithm and available information about the user's needs in order to recommend to him a relevant set of objects that are most useful to him.

An analysis of recent research and publications in the field of introducing recommender systems has shown that improving the quality of recommender system proposals based on machine learning methods is an urgent task. The use of neural networks in recommender systems can improve the efficiency and usability of these systems.

The aim of the study is to improve the quality of recommender system proposals based on machine learning methods.

In the course of the study, a set of stages was systematized and a methodology for building an effective recommender system using machine learning methods was determined. The mechanisms are defined, the stages are formalized, and a technical block diagram for the development of a neural recommender system is presented. Two models of neural network recommender systems are built using joint filtering and deep matrix factorization.

Evaluation of recommendation systems by indicators was carried out Precision, Recall and Normalized Discounted Cumulative Gain.

Research was carried out using such optimization algorithms: SGD, RMSprop, ADAdelta and FTRL. The results of an experimental study of deep neural network models when generating recommendations in various scenarios showed that their performance can vary greatly depending on the search model, the amount and quality of data, as well as the network architecture and network training method.

Based on the results of the experiments, the optimal learning algorithms for the neural network recommender system model were determined for solving a specific problem, depending on the nature of the initial data.

The experimental research was conducted using Python and TensorFlow.

The work uses freely available datasets on movie ratings from the MovieLens website.

Keywords: *recommendation system, filtering methods, machine learning algorithms, performance evaluation.*

Шумова Л.О. – к.т.н., доцент, доцент кафедри комп’ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: shumova@snu.edu.ua

Рязанцев О.І. – д.т.н., професор, завідувач кафедри комп’ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: a_ryazantsev@snu.edu.ua

Покришка С.А. – аспірант кафедри комп’ютерних наук та інженерії Східноукраїнського національного університету імені Володимира Даля, e-mail: hakermans@gmail.com

Стаття подана 23.04.2023.