РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДВУХБАШЕННОЙ АРХИТЕКТРЫ

А. В. Сергеев

Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь, akvise@gmail.com
Научный руководитель – В. И. Малюгин, доктор экономических наук, доцент

Данная статья посвящена разработке модели персонализированной рекомендательной системы с двухбашенной архитектурой на основе контентных признаков.

Ключевые слова: рекомендательные системы, персонализированные рекомендации, двухбашенная архитектура, машинное обучение, глубокое обучение, PyTorch.

1. Введение в рекомендательные системы, цели исследования

Рекомендательные системы стали ключевым элементом в сфере информационных технологий, изменяя способы взаимодействия пользователей с контентом. Их развитие началось с конкурса Netflix в 2006 году, где за улучшение алгоритма рекомендаций был предложен приз в миллион долларов. Это событие стимулировало появление новых методов, в частности, матричной факторизации и алгоритмов SVD++. С ростом интернета и объемов информации, рекомендательные системы нашли применение в различных областях, таких как электронная коммерция, социальные сети и стриминговые сервисы, что способствовало бурному развитию технологий в этой сфере.

В работе решаются следующие основные задачи:

1. Подготовка обзора по принципам организации и методам построения рекомендательных систем для моделей с двухбашенной архитектурой.
2. Разработка общей схемы алгоритма, реализующего модель рекомендательной системы на основе двухбашенной архитектуры с применением нейронных сетей.
3. Проведение экспериментальных исследований на реальных данных предложенной модели для оптимизации и актуализации параметров нейронной сети.
4. Разработка программы для обучения нейронной сети, реализующей рекомендательную систему с двухбашенной архитектурой.

2. Описание архитектуры двухбашенной модели

В моей работе рассматривается реализация двухбашенной модели с использованием библиотеки PyTorch. Модель включает две независимые нейронные сети: одна обрабатывает данные пользователей, другая — данные элементов (например, фильмов). Каждая сеть преобразует входные данные в

эмбеддинги (векторное представление чего-либо) фиксированного размера, которые затем объединяются для предсказания рейтинга, который пользователь может поставить элементу.

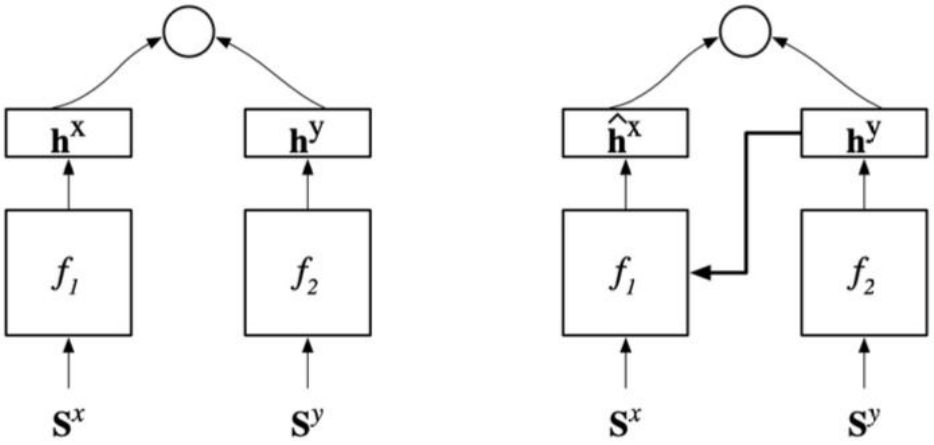

Рис. 1. Пример архитектура DSSM

Класс UserTower создает эмбеддинги пользователей и обрабатывает их через три полносвязных слоя с функцией активации ReLU, что позволяет уловить сложные зависимости в данных. Аналогично, класс ItemTower преобразует идентификаторы элементов в эмбеддинги и пропускает их через три слоя с ReLU, кодируя значимые характеристики элементов. Объединенная модель извлекает эмбеддинги пользователей и элементов, вычисляя их скалярное произведение для предсказания рейтинга. Считаем функцию потерь, которая оценивает разницу между предсказанными и реальными рейтингами, а оптимизатор Adam обеспечивает эффективное обновление весов модели, улучшая точность предсказаний.

3. Подбор гиперпараметров

Гиперпараметры — это характеристики модели, которые фиксируются до начала обучения, такие как глубина решающего дерева, значение силы регуляризации в линейной модели и learning rate для градиентного спуска. В отличие от параметров модели, которые настраиваются в процессе обучения, гиперпараметры задаются разработчиком заранее. Optuna — это фреймворк для автоматизированного поиска оптимальных гиперпараметров для моделей машинного обучения, который подбирает эти параметры методом проб и ошибок.

Ключевые особенности Optuna:

1. *Настраиваемое пространство поиска гиперпараметров:* Разработчик может самостоятельно задать пространство для поиска гиперпараметров, используя базовый синтаксис Python (циклы, условия).

2. *Алгоритмы SoTA для выбора гиперпараметров и ранней остановки*: Optuna предоставляет различные алгоритмы для семплирования гиперпараметров (samplers) и прунинга (pruners), позволяя разработчику выбрать дефолтный алгоритм, использовать конкретный или создать свой собственный.
3. *Легкость распараллеливания процесса поиска гиперпараметров*: Optuna поддерживает параллельный поиск гиперпараметров и может быть дополнена dashboard для визуализации обучения в реальном времени.

4. Обучение модели

Оптимизация гиперпараметров позволила выявить, что наиболее значимыми параметрами для улучшения производительности модели оказались размер эмбеддингов и скорость обучения. Наилучшие результаты были достигнуты при следующих значениях гиперпараметров:

1. Размер эмбеддингов: 32.
2. Количество нейронов в скрытых слоях: 64.
3. Скорость обучения: 0.001.

После завершения обучения модели с двухбашенной архитектурой и логирования метрик в WandB, я приступил к оценке модели на проверочной выборке. Для этого я загрузил сохраненную модель и прогнал проверочные данные через нее. Метрика NDCG (Normalized Discounted Cumulative Gain) измеряет качество ранжирования элементов по их релевантности. Она широко используется в задачах рекомендательных систем, где важно правильно упорядочить рекомендации для пользователя.

Метрика NDCG вычисляется путем суммирования релевантностей элементов с учетом их позиции в ранжированном списке и применения дисконтирования. В данном случае, метрика NDCG@10 означает, что мы оцениваем качество ранжирования для первых 10 рекомендаций за сессию пользователя.

Финальные метрики были получены после нескольких итераций улучшения модели. Была выбрана модель, которая показывала наилучшую производительность в ранжировании рекомендаций по метрике NDCG@10. Эти финальные метрики отражали качество и эффективность модели в решении задачи рекомендации.

В работе [7] сравнили лучшую версию DSSM (строка 12) с тремя наборами базовых моделей. Первый набор включает методы лексического сопоставления, такие как TF-IDF (строка 1) и BM25 (строка 2). Второй набор представлен моделью перевода слов (WTM в строке 3), предназначенной для решения проблемы несоответствия языка запроса и документа

путем изучения лексического соответствия между словами запроса и документа. Третий набор включает современные латентные семантические модели, изучаемые на документах либо в неконтролируемом режиме (LSA, PLSA, DAE, строки 4-6), либо в контролируемом режиме (BLTM-PR, DPM, строки 7-8). Для сопоставимости результатов мы переопределили эти модели, обучив LSA и DPM с использованием словаря из 40 тысяч слов, а остальные модели — с использованием словаря из 500 тысяч слов. Полученные результаты анализа для различных метрик представлены в таблице на рис. 12

#	Models	NDCG@1	NDCG@3	NDCG@10
1	TF-IDF	0.319	0.382	0.462
2	BM25	0.308	0.373	0.455
3	WTM	0.332	0.400	0.478
4	LSA	0.298	0.372	0.455
5	PLSA	0.295	0.371	0.456
6	DAE	0.310	0.377	0.459
7	BLTM-PR	0.337	0.403	0.480
8	DPM	0.329	0.401	0.479
9	DNN	0.342	0.410	0.486
10	L-WH linear	0.357	0.422	0.495
11	L-WH non-linear	0.357	0.421	0.494
12	**L-WH DNN**	**0.362**	**0.425**	**0.498**

Рис. 2. Результаты анализа для различных метрик

Заключение

В работе освещаются различные архитектуры рекомендательных систем, включая алгоритмы с основным вниманием на двухбашенной архитектуре. Анализируются достоинства и недостатки этой архитектуры, а также охватываются эксперименты, которые включают весь процесс построения рекомендательной системы от предобработки данных до обучения модели и её оптимизации по гиперпараметрам. В результате был выбран окончательный вариант модели, продемонстрировавший наилучшее качество среди других моделей.

Библиографические ссылки

1. Koren Y. Matrix factorization techniques for recommender systems / Y. Koren, R. Bell, C. Volinsky // Computer. No42. 2009. P. 30-37.

2. Sarwar B. Item-based collaborative filtering recommendation algorithms / B. Sarwar [et al.] // Proceedings of the 10th International Conference on World Wide Web. 2001. P. 285-295.

3. Herlocker J. L. Evaluating collaborative filtering recommender systems / J. L. Herlocker [et al.] // ACM Transactions on Information Systems. No 22. 2004. P. 5-53.

4. Kumar, V. Deep Neural Architecture for News Recommendation [Electronic resource] - Mode of access: https://ceur-ws.org/Vol-1866/paper_85.pdf. - Date of access: 23.04.2024

5. Adomavicius, G. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions / G. Adomavicius, A. Tuzhilin // IEEE Transactions on Knowledge and Data Engineering. No 17. 2005. P. 734- 749.

6. Zhang Y. Deep learning for recommender systems: A rigorous survey. arXiv preprint arXiv:1707.07435. 2016.

7. Huang P. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data [Electronic resource] - Mode of access: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/cikm2013_DSSM_fullversion.pdf. - Date of access: 2013