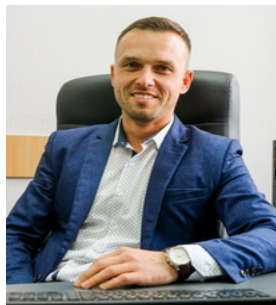


УДК 004.021:004.832.2

## МОДЕЛИРОВАНИЕ ПЕРСОНАЛИЗИРОВАННОЙ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ



**М.Д. Михальков**  
магистрант кафедры  
ПОИТ, ФКСиС, БГУИР  
solution.m@mail.ru



**С.Н. Нестеренков**  
кандидат технических наук,  
доцент кафедры ПОИТ,  
ФКСиС, БГУИР  
s.nesterenkov@bsuir.by

### **М.Д. Михальков**

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с анализом и разработкой моделей и алгоритмов построения интеллектуальных систем поддержки принятия решений.

### **С.Н. Нестеренков**

Окончил аспирантуру Белорусского государственного университета информатики и радиоэлектроники (2013), в 2017 защитил диссертацию на соискание ученой степени кандидата технических наук по специальности 05.13.01 - «Системный анализ, управление и обработка информации». Является доцентом кафедры ПОИТ.

**Аннотация.** В рамках данного исследования было проведено моделирование рекомендательной системы, предназначенной для платформ электронной коммерции. В процессе моделирования были рассмотрены различные архитектурные решения, формализованы основные компоненты системы и разработаны математические модели, описывающие работу рекомендательных алгоритмов.

Особое внимание было уделено выбору и анализу методов рекомендаций, включая гибридные модели и глубокое обучение. В данной статье было представлено концептуальное моделирование системы, описание ее логической и функциональной структуры, выбор формализованных методов и алгоритмов, а также анализ возможных сценариев работы.

**Ключевые слова:** Рекомендательные системы, глубокое обучение, DLRM, матричная факторизация, гибридная фильтрация.

**Введение.** В статьях [1, 2] был представлен обзор моделей и методов построения интеллектуальных систем поддержки принятия решений в области электронной коммерции. В данном же исследовании будет рассмотрено описание модели рекомендательной системы, включая основные подходы и методы, используемые для создания эффективных рекомендаций. Подразумевается, что в последующем будет смоделирована общая система, которая представляет собой рекомендательную систему, интегрированную с интеллектуальной системой поддержки принятия решений.

Глубокое обучение – это современное решение многих проблем машинного обучения, которое превосходит альтернативные методы. Последние тенденции включают применение методов глубокого обучения в рекомендательных системах. Многие крупные компании, такие как AirBnB, Facebook, Google, Home Depot, LinkedIn и Pinterest, делятся своим опытом использования глубокого обучения для рекомендательных систем.

Далее будет рассмотрена архитектура моделей на основе глубокого обучения.

**Описание базовой модели рекомендательной системы на основе глубокого обучения.** Эмбединг-слои представляют категории с плотными векторами. Эта техника популярна в *NLP* для встраивания слов с плотным представлением. Слова с похожим значением имеют похожий эмбединг-вектор. Эту же технику можно применить к рекомендательным системам. Самая тривиальная рекомендательная система основана на пользователях и объектах-продуктах (айтемах). Имеются идентификаторы пользователей и идентификаторы айтемов. Слова – это пользователи и айтемы, поэтому в данном случае рассматривается две эмбединг-таблицы (рисунок 1).

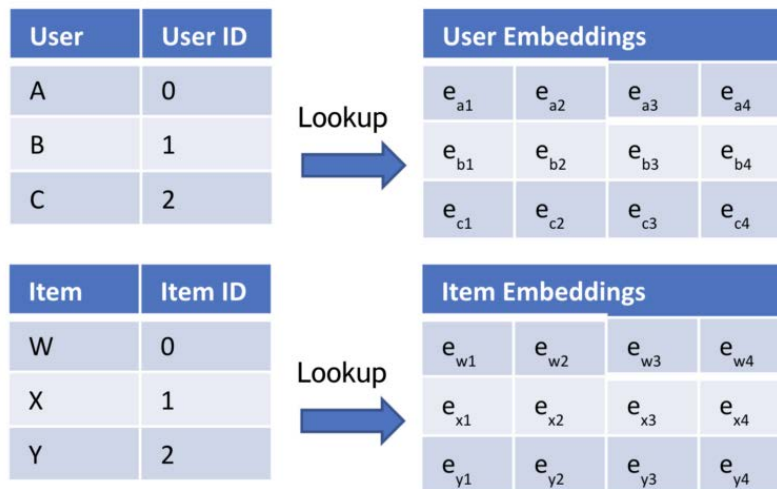


Рисунок 1. Эмбединг-таблицы пользователей и айтемов

Вычисление *dot*-умножения между эмбедингом пользователя и эмбедингом айтема позволяет получить окончательный результат – оценку релевантности данного айтема для пользователя. Этот подход широко применяется в нейросетевых рекомендательных системах, так как он эффективно моделирует скрытые предпочтения пользователя.

Для получения вероятности взаимодействия данный результат можно пропустить через сигмоидную функцию активации, которая преобразует значение в диапазон от 0 до 1. Это особенно полезно в бинарных задачах предсказания, таких как определение вероятности клика или покупки.

Дополнительно, для улучшения качества рекомендаций можно применять нормализацию эмбедингов, регуляризацию или обучение с учетом отрицательных примеров. На рисунке 2 представлена схема этого процесса.

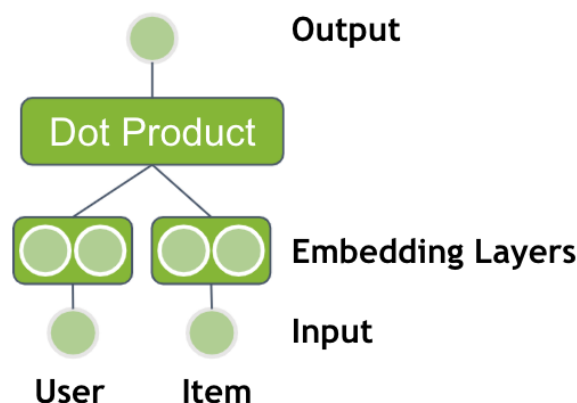


Рисунок 2. Нейронная сеть с двумя эмбединг-таблицами и *dot*-продуктом на выходе

Этот метод эквивалентен матричной факторизации или методу наименьших чередующихся квадратов (*Alternating Least Squares, ALS*).

Однако производительность нейронных сетей основана на более глубоких архитектурах с несколькими нелинейными слоями. Можно расширить предыдущую модель, подавая выход эмбединг-слоев через несколько полносвязных слоев с активациями *ReLU*.

Выбор дизайна заключается в том, как объединить два эмбединг-вектора. Можно либо объединить только эмбединг-векторы, либо поэлементно умножить векторы друг на друга, по аналогии с *dot*-продуктом. За выводом следуют несколько скрытых слоев (рисунок 3).

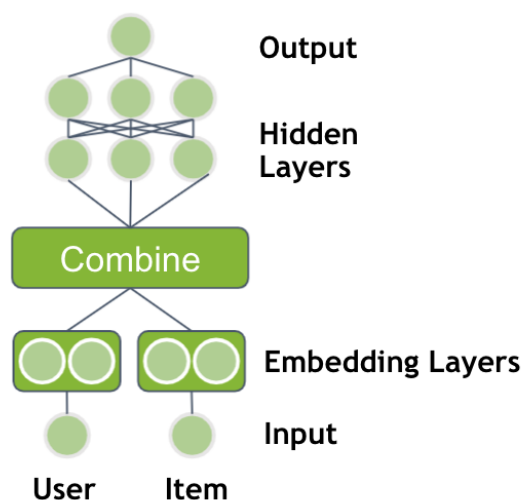


Рисунок 3. Нейронная сеть с несколькими полносвязными слоями

До сих пор в качестве входных данных использовались только идентификатор пользователя и идентификатор продукта, но зачастую в распоряжении имеется гораздо больше информации. Дополнительной информацией о пользователе может быть пол, возраст, город (адрес), время с момента последнего посещения или кредитная карта, использованная для оплаты. У продукта-товара обычно есть бренд, цена, категории или количество, проданное за последние 7 дней. Эта сопутствующая информация может помочь модели более успешно выполнять обобщение. Модифицировав нейронную сеть, можно использовать дополнительные характеристики в качестве входных данных (рисунок 4).

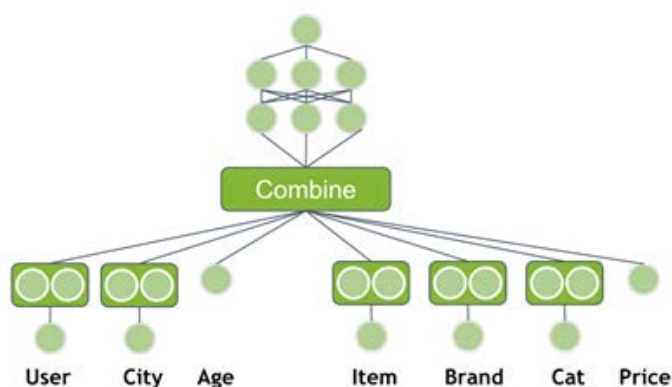


Рисунок 4. Нейронная сеть с метаинформацией и несколькими полносвязными слоями

Базовой моделью для разрабатываемой рекомендательной системы на основе глубокого обучения была выбрана *DLRM* от *Facebook* [3]. Она имеет структуру, аналогичную архитектуре нейронной сети с метаданными (рисунок 4), но обладает некоторыми специфическими отличиями (рисунок 5).

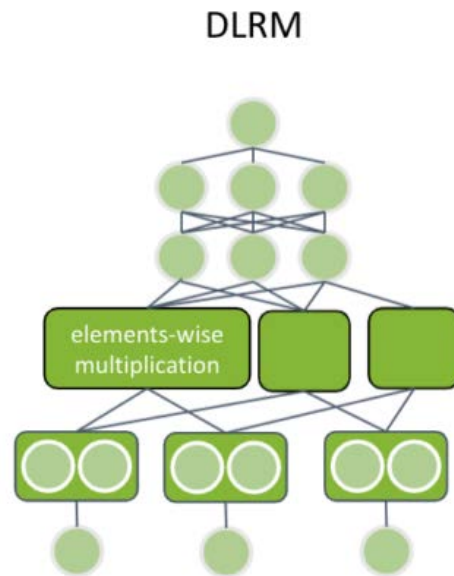


Рисунок 5. Архитектура *DLRM*

Важным аспектом исследования является рассмотрение рабочего процесса модели *DLRM*. Модель использует встраивания для обработки разреженных признаков, которые представляют категориальные данные, и многоуровневые перцептроны (*MLP*) для обработки плотных признаков. Эти признаки взаимодействуют с помощью статистических методов, предложенных в модели. В конечном итоге она находит вероятность события, выполняя пост-обработку взаимодействий с другим *MLP* (рисунок 6).

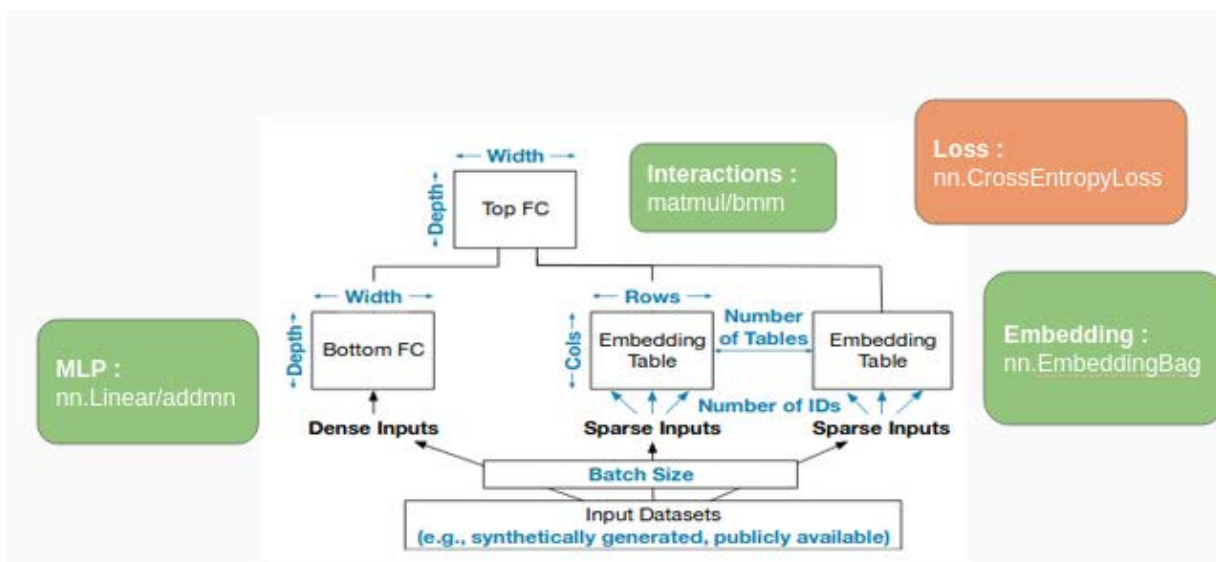


Рисунок 6. Рабочий процесс модели *DLRM*

В архитектуре модели можно выделить следующие основные аспекты: встраивания, матрица факторизации, машины факторизации и многоуровневый перцептрон (*MLP*).

Встраивание – это отображение концепций, объектов или элементов в векторное пространство. В контексте нейронных сетей встраивания – это низкоразмерные, обучаемые,

непрерывные векторные представления дискретных переменных. Они позволяют моделям машинного обучения выявлять скрытые закономерности и семантические связи между объектами.

Они широко применяются в областях, где необходимо преобразовать дискретные категории в числовые представления для последующего анализа и обработки. В рекомендательных системах встраивания используются для представления пользователей и товаров в общем пространстве, что позволяет находить скрытые предпочтения и предлагать более точные рекомендации.

В чем стоит преимущество использования встраивания вместо других вариантов, таких как списки разреженных элементов? Встраивания уменьшает размерность категориальных переменных и осмысленно представляет категории в абстрактном пространстве. Можно измерять расстояние между встраиваниями более осмысленным способом. Встраивания элементов представляют разреженные признаки в некотором абстрактном пространстве, релевантном для модели, в то время как целые числа представляют упорядочивание входных данных. Вектор встраивания проецирует пространство  $n$ -мерных элементов в  $d$ -мерные векторы встраивания, где  $n \gg d$ .

Пример встраивания представлен на рисунке 7.

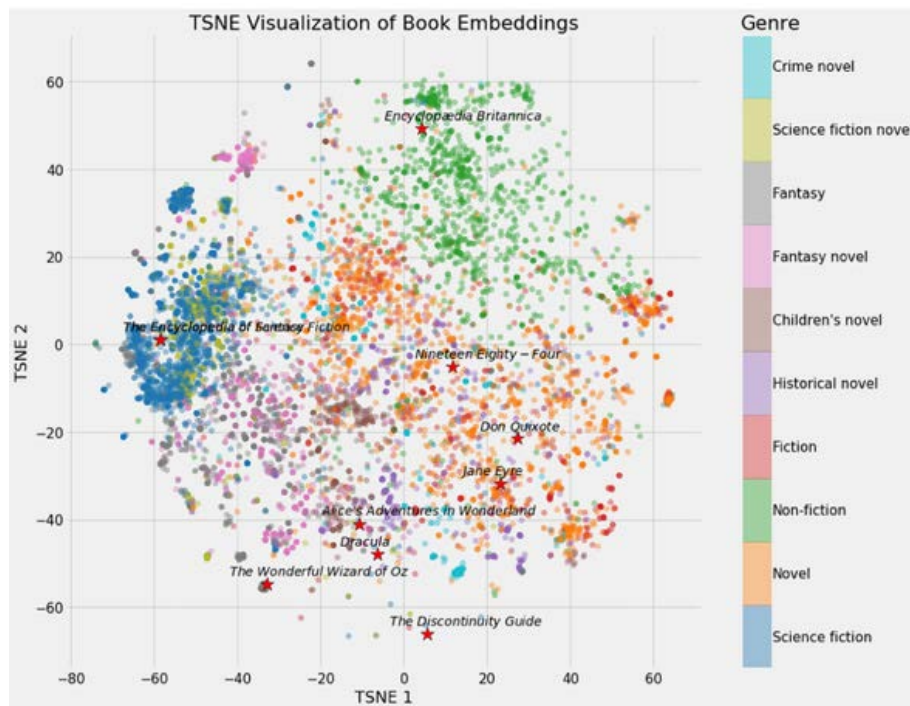


Рисунок 7. Визуализация встраивания

Матрица факторизации – это техника, которая принадлежит к классу алгоритмов совместной фильтрации, используемых в системах рекомендаций. Она позволяет выявлять скрытые паттерны взаимодействий пользователей с элементами, что делает рекомендации более точными и персонализированными.

Алгоритмы матричной факторизации работают, разлагая матрицу взаимодействий пользователей и элементов на произведение двух матриц меньшей размерности (рисунок 8). Эти матрицы представляют собой эмбединги пользователей и элементов в общем латентном пространстве признаков. Взаимодействие между пользователем и элементом можно оценить с помощью скалярного произведения их эмбедингов.

Несмотря на свою эффективность, матричная факторизация имеет ограничения, например, сложность работы с разреженными и динамически изменяющимися данными.



Для их преодоления современные рекомендательные системы часто дополняют матричные разложения методами глубокого обучения и гибридными моделями.

Машины факторизации (*FM*) – это улучшенная версия матричной факторизации. На самом деле *FM* можно рассматривать как обобщение нескольких классических методов машинного обучения. При определенных параметрах они могут вести себя как линейные модели, матричная факторизация или полиномиальные модели, что делает их гибким и мощным инструментом для анализа данных. Машины факторизации предназначены для эффективного захвата взаимодействий между признаками в высокоразмерных разреженных данных. Они способны оценивать взаимодействия в разреженных условиях, поскольку они нарушают независимость взаимодействий между параметрами, факторизуя их. Кроме того, машины факторизации включают взаимодействия второго порядка в линейную модель с категориальными данными, определяя модель в следующем виде, рисунок 9.

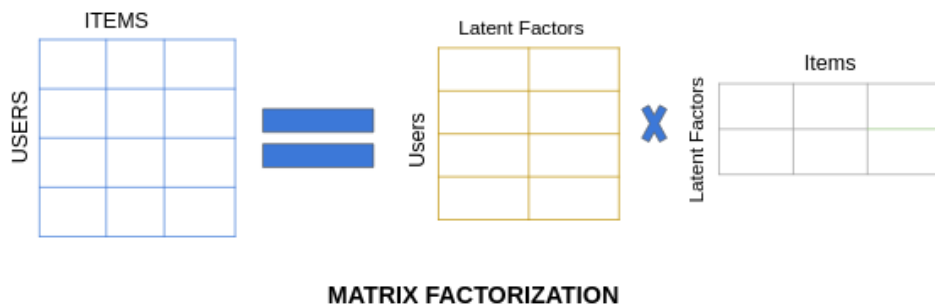


Рисунок 8. Визуализация встраивания

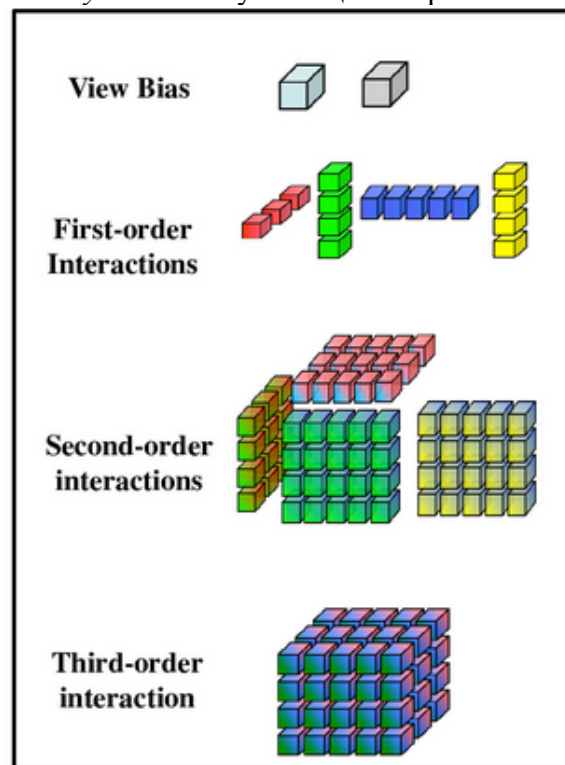


Рисунок 9. Различные порядки матриц взаимодействия

*FM* факторизуют матрицу взаимодействий второго порядка в ее латентные факторы (или вектора встраивания), как в матричной факторизации, что более эффективно обрабатывает разреженные данные. Немаловажно, что они значительно снижают

сложность взаимодействий второго порядка, захватывая только взаимодействия между парами различных векторов встраивания, что дает линейную вычислительную сложность.

Многоуровневый перцептрон (*MLP*) – это класс нейронных сетей с прямой связью, состоящих как минимум из трех слоев узлов: входного слоя, скрытого слоя и выходного слоя. Все узлы, кроме входных, являются нейронами, которые используют нелинейные активационные функции.

Эти методы были использованы для захвата более сложных взаимодействий. *MLP* с достаточной глубиной и шириной могут подгонять данные с произвольной точностью.

**Описание улучшенной модели персонализированной рекомендательной системы.** Моделирование рекомендательной системы с использованием глубокого обучения имеет в своей основе решение задачи предсказания предпочтений пользователей на основе их истории взаимодействий с системой. В данной статье модель *DLRM* будет модифицирована для решения задач с использованием регуляризации, оптимизации вычислений, гибридной фильтрации и учета дополнительных признаков пользователей и товаров. Все эти улучшения направлены на повышение эффективности персонализированных рекомендаций, а также адаптации работы рекомендательной системы.

Первое улучшение модели *DLRM* состоит в стабилизации процесса обучения. Для этого используются методы *Dropout* и *Batch Normalization (BatchNorm)*. Это методы регуляризации, которые помогают улучшить обобщающие способности модели.

*Dropout* заключается в случайном исключении некоторых нейронов на каждом шаге обучения, что предотвращает переобучение, улучшает обобщение и помогает бороться с избыточной адаптацией модели к данным. Математически, *Dropout* можно представить как операцию, которая умножает выход нейрона на случайную бинарную маску  $m$  размерности  $n$ , где  $n$  – количество нейронов, формула (1):

$$y^{drop} = m \odot y, \quad (1)$$

где  $y$  – выход нейрона;  $m$  – маска;  $\odot$  – операция покомпонентного умножения.

*Batch Normalization (BatchNorm)* нормализует выходы слоев нейронной сети на каждом шаге обучения. Она вычисляет среднее и стандартное отклонение на батче данных и использует их для нормализации каждого слоя, формула (2):

$$\hat{x} = \frac{x - \mu_B}{\sigma_B}, \quad (2)$$

где  $\mu_B$  – среднее по батчу;  $\sigma_B$  – стандартное отклонение;  $x$  – входные данные слоя.

После нормализации данные могут быть масштабированы с помощью обучаемых параметров  $\gamma$  и  $\beta$ , формула (3):

$$y = \gamma \hat{x} + \beta. \quad (3)$$

Это помогает ускорить обучение и повышает стабильность, уменьшив проблему с исчезающими или взрывающимися градиентами.

В контексте рекомендательных систем краеугольным камнем является вопрос оптимизации времени предсказаний. Для этих целей будет использован *Quantization* – это метод преобразования числовых значений в модели с более высокой точностью в представление с меньшей точностью (например, 32-битное представление заменяется 8-битным). В математическом выражении для квантования можно записать следующее преобразование, формула (4):

$$Q(x) = \text{round}\left(\frac{x}{\Delta}\right) * \Delta, \quad (4)$$

где  $\Delta$  – шаг квантования;  $x$  – исходное значение;  $\text{round}(\cdot)$  – операция округления.

Это позволяет уменьшить размер модели и ускорить вычисления, не значительно ухудшая точность предсказания. В контексте рекомендательных систем квантование может быть использовано для ускорения инференса, особенно при обработке больших данных и работе в реальном времени.

Оптимизация модели производится через *TorchScript* и *TensorRT*.

*TorchScript* позволяет превращать модель, написанную в *PyTorch*, в статическую форму, которую можно оптимизировать и выполнять без *Python*. В этом случае модель представляется как вычислительный граф, который можно оптимизировать для быстрого выполнения.

Математически, *TorchScript* преобразует исходный динамический граф *PyTorch* в статический граф вычислений  $G(\theta)$ , где  $\theta$  – параметры модели. Преобразование в статический граф позволяет использовать такие оптимизации, как удаление лишних операций, слияние слоев и других техник, которые ускоряют инференс.

*TensorRT* представляет собой библиотеку для оптимизации выполнения нейронных сетей на графических процессорах (GPU) от NVIDIA. *TensorRT* использует оптимизации, такие как слияние слоев, использование пониженной точности (FP16, INT8), аппаратная оптимизация для GPU. Это позволяет ускорить выполнение операций свертки, умножения матриц и других типичных операций в нейронных сетях.

Для улучшения точности рекомендаций важным аспектом является добавление дополнительных признаков, таких как история покупок или поведение на сайте. Это позволяет дополнительно персонализировать рекомендации, учитывая не только текущие предпочтения пользователя, но и его поведение в прошлом.

Категориальные эмбединги для пользователя и товара позволяют эффективно представлять их характеристики в виде низкоразмерных векторов, что значительно сокращает размер пространства признаков.

Вместо классической коллаборативной фильтрации, которая реализуется в базовой модели *DLRM*, в улучшенной модели персонализированной рекомендательной системы будет использоваться гибридная фильтрация в совокупности с матричной факторизацией. Матричная факторизация является одним из ключевых методов и заключается в разложении матрицы взаимодействий пользователей и товаров  $R$  на две низкоразмерные матрицы, формула (5):

$$R \approx P * Q^T, \quad (5)$$

где  $P$  – матрица скрытых факторов пользователей;  $Q$  – матрица скрытых факторов товаров.

Эти матрицы обучаются таким образом, чтобы минимизировать ошибку предсказания, формула (6):

$$\min_{P, Q} \sum_{i, j} (r_{ij} - P_i Q_j^T)^2, \quad (6)$$

где  $r_{ij}$  – оценка взаимодействия;  $P_i$  и  $Q_j$  – скрытые факторы для пользователя  $i$  и товара  $j$ .

**Заключение.** Таким образом, математическое описание улучшенной модели персонализированной рекомендательной системы на основе *DLRM* включает в себя широкий набор методов и техник, таких как регуляризация (*Dropout*, *BatchNorm*), оптимизация вычислений (*Quantization*, *TorchScript*, *TensorRT*), использование эмбедингов для представления категориальных признаков и гибридная фильтрация с



матричной факторизацией. Эти методы в совокупности обеспечивают высокую точность, скорость и стабильность работы модели, что особенно важно при обработке больших и сложных данных в реальных задачах рекомендательных систем.

### Список литературы

[1] Михальков, М. Д. Преимущества и недостатки эмпирических моделей интеллектуальных систем поддержки принятия решений в области электронной коммерции = Advantages and disadvantages of empirical models of intelligent decision support systems in the field of e-commerce / М. Д. Михальков // BIG DATA и анализ высокого уровня = BIG DATA and Advanced Analytics : сборник научных статей X Международной научно-практической конференции, Минск, 13 марта 2024 г. : в 2 ч. Ч. 2 / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: В. А. Богуш [и др.]. – Минск, 2024. – С. 201–207.

[2] Михальков, М. Д. Модели и методы построения интеллектуальных систем поддержки принятия решений / М. Д. Михальков // Информационные технологии и системы 2023 (ИТС 2023) = Information Technologies and Systems 2023 (ITS 2023) : материалы Международной научной конференции, Минск, 22 ноября 2023 / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2023. – С. 67–68.

[3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In Proc. 2nd Conference on Systems and Machine Learning (SysML), 2019.

### Авторский вклад

**Михальков Максим Дмитриевич** – моделирование рекомендательной системы, предназначенной для платформ электронной коммерции, рассмотрение различных архитектурных решений, формализация основных компонентов системы и разработка математических моделей, описывающих работу рекомендательных алгоритмов.

**Нестеренков Сергей Николаевич** – постановка задачи исследования, анализ полученных результатов.

## MODELING A PERSONALIZED RECOMMENDATION SYSTEM BASED ON DEEP LEARNING

**M.D. Mikhalkov**

*PhD student at the Department of  
Information Technology Software,  
Faculty of Computer Systems and  
Networks, BSUIR*

**S.N. Nesterenkov**

*PhD of Technical Sciences,  
Associate Professor at the  
Department of Information  
Technology Software, Faculty of  
Computer Systems and Networks,  
BSUIR*

**Abstract.** In this article, a modeling of a recommender system designed for e-commerce platforms was conducted. During the modeling process, various architectural solutions were considered, the main components of the system were formalized, and mathematical models describing the operation of recommender algorithms were developed.

Particular attention was paid to the selection and analysis of recommendation methods, including hybrid models and deep learning. This article presented a conceptual modeling of the system, a description of its logical and functional structure, a selection of formalized methods and algorithms, and an analysis of possible operating scenarios.

**Keywords:** Recommender systems, deep learning, DLRM, matrix factorization, hybrid filtering.