Faculty of Engineering & Technology

Department of Electrical & Computer Engineering

Accelerated DLRM-based E-commerce Recommendation System

Prepared By:

Ibraheem Alyan 1201180

Mohammad Abu-Shelbaia 1200198

Nidal Zabade 1200153

Supervised By:

Dr. Ahmed Shawahna

A graduation project report submitted to Electrical and Computer Engineering Department as part of B.Sc. in Computer Engineering degree requirements fulfilments.

Birzeit

December 20, 2023

Abstract

The project aims to design and develop a cutting-edge accelerated e-commerce deep learning recommendation system. The goal is to deliver a production-ready solution, with automated data injestion and training pipelines, and a simple RESTful API as a final interface. The project will have special focus on scalabilty and performance.

This report discusess different types of recommendation systems and compares them to DLRM based systems in terms of different metrics and features. Furthermore, it compares existing solutions and their aspects, and discuesses possible technolgies and architectures to use in the system.

المستخلص

يهدف المشروع إلى تصميم وتطوير نظام توصية لمنصات التجارة الإلكترونية باستعمال التعلم الآلي العميق. الهدف النهائي هو تقديم حلول صالحة لبيئة التشغيل، تتم فيها أتمتة عمليات إدخال البيانات و تدريب نماذج التعلم اللآلي وواجهة برمجة تطبيقات غيصةفل اعي كواجهة نهائية. سيركز المشروع بشكل خاص على قابلية التوسع والأداء.

يناقش هذا التقرير أنواعًا مختلفة من أنظمة التوصيات ويقارنها بالأنظمة القائمة على نماذج التوصية بالتعلم اللآلي العميق (ضغپ) من حيث المقاييس والميزات المختلفة. علاوة على ذلك، فهو يقارن الحلول المتوفرة حالياً و مزاياها، كما ويناقش التقنيات والبنى الممكن استعمالها في تطوير النظام.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Motivation

The exponential growth of e-commerce has introduced an enormous amount of choice, where consumers face overwhelming product options. To address this challenge, personalized recommendation systems have become essential for enhancing the shopping experience, and increasing the conversion rate for any e-commerce platform.

In contrast to conventional collaborative filtering[6], content-based[6], or popularity-based recommendation systems, our AI-based solution offers distinct advantages. Firstly, AI makes it possible to provide per-user personalized recommendations, which are tailored to their unique preferences and behaviors, enhancing user engagement and satisfaction. AI systems can also intelligently recommend comparable or complementary products or content to increase revenue through cross-selling. Furthermore, AI takes into account the impressions and interactions of users with items, allowing for a more dynamic and accurate understanding of user preferences. Using AI leads to improved recommendation accuracy and relevancy, leading to increased conversion rates and business growth.

Statistics from different use cases of recommendation systems:

- On average, an intelligent recommender system delivers a 22.66% lift in conversions rates [7] for web products.
- IKEA experienced a 30% increase in click-through rate, 2% surge in average order value [8] using Google Recommendations AI [4]
- Lotte Mart increased new product purchases by 1.7x [9] using Amazon Personalize [3]

In summary, our project's motivation is elevating the e-commerce experience, driving business success, and harnessing cutting-edge AI technologies to create a recommendation system that is both high-performing and scalable.

1.2 Problem Statement

the process of building the solution is mainly two parts:

- First, designing a personalized recommendation system that covers what traditional collaborative filtering, content-based, or popularity-based systems cannot achieve.
- Second, deploying and automating the solution, including, data cleaning, data storage, and model deployment processes, and ensuring a production-ready and scalable system.

Chapter 2

Background

Contents

2.1 Recommendation Systems

"A recommendation system is an artificial intelligence or AI algorithm, usually associated with machine learning, that uses Big Data to suggest or recommend additional products to consumers. These can be based on various criteria, including past purchases, search history, demographic information, and other factors."[6]

Recommender systems undergo training to understand the preferences, earlier decisions, and attributes of the user and products using their past interactions which includes impressions, clicks, purchases, and ratings. Recommender systems are usually used by content and product providers to suggest items to users that they may like based on their profile and preferences.

2.2 Types of Recommendation Systems

2.2.1 Context Filtering

Context Filtering is a technique that uses the contextual information of the user by framing the recommendation problem as a contextual multi-armed bandit problem and using the contextual information to learn the user's preferences.

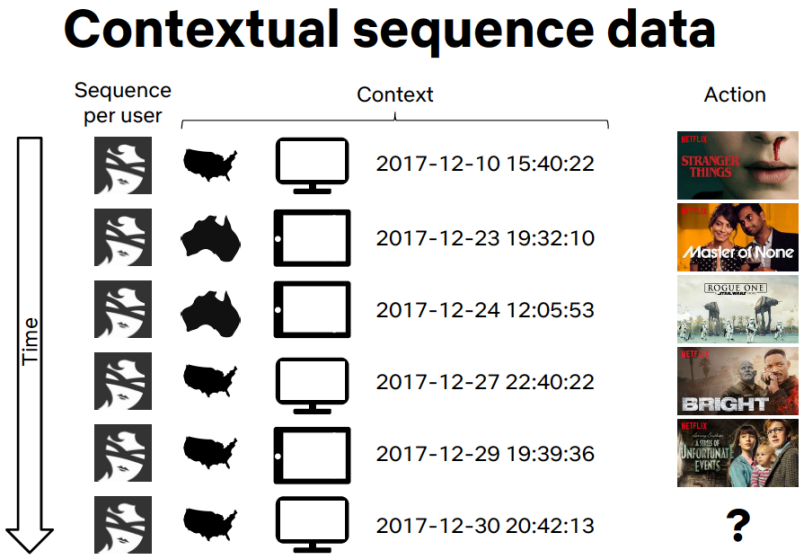


Figure 2.1: Context Filtering Diagram
[6]

2.2.2 Variational Autoencoder for Collaborative Filtering

This model consists of two parts: an encoder and a decoder. The encoder takes the user's preferences as input and encodes them into a latent space. The decoder takes the latent space as input and decodes it into the item's features.

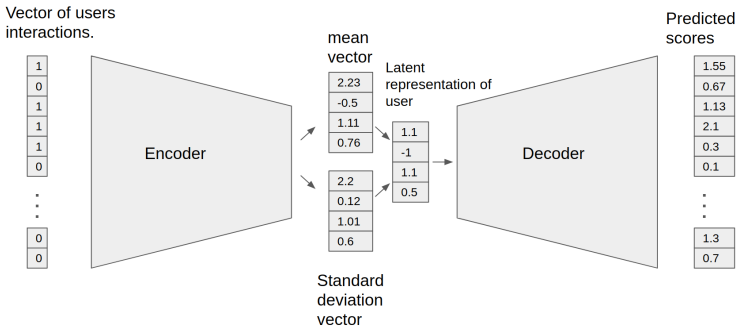


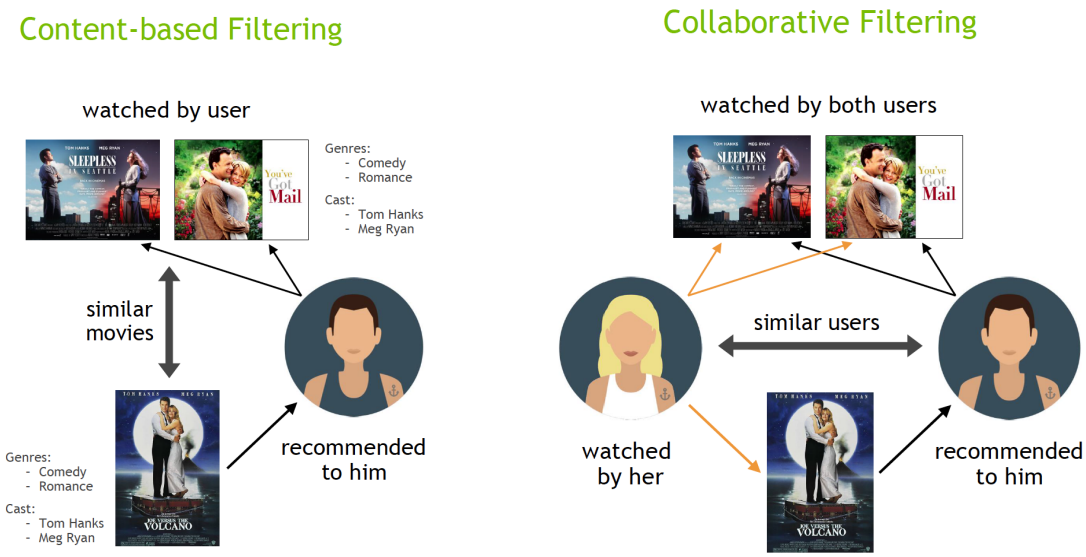
Figure 2.2: Variational Autoencoder for Collaborative Filtering Structure
[6]

2.2.3 Collaborative Filtering

Collaborative Filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. It looks at the items they like and combines them to create a ranked list of suggestions. This technique is based on the idea that people who agreed in the past will agree in the future.

2.2.4 Content Filtering

Content Filtering is a technique that uses the features of items a user has interacted with to recommend more items with similar features. This technique is based on the idea that if a customer showed interest in a particular product, he will also be interested in a similar product.



(a) Content Filtering (b) Collaborative Filtering Diagram

Figure 2.3: Nvidia Golssary Diagram[6]

2.2.5 Hybrid Recommendation Systems

Hybrid Recommendation Systems combine the advantages of multiple types of recommendation algorithms to create a more comprehensive recommending system.

2.2.6 Neural Collaborative Filtering

A technique that uses neural networks to learn the customer's preferences and recommend items. It uses a neural network to learn the user's preferences and a neural network to learn the item's features. The two networks are then combined to create a recommendation.

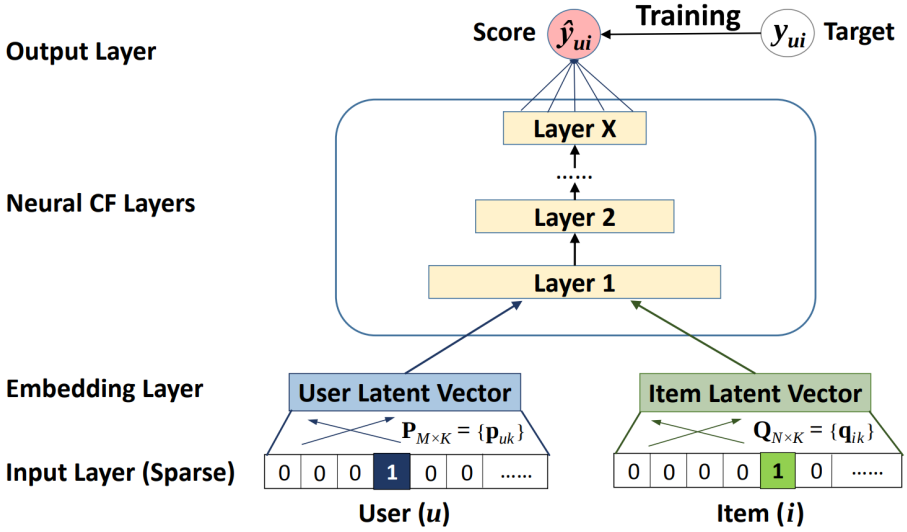


Figure 2.4: Neural Collaborative Filtering
[6]

2.2.7 Contextual Sequence Learning

Contextual Sequence Learning is a technique that uses a recurrent neural network to learn the user's preferences and a neural network to learn the item's features. The two networks are then combined to create a recommendation.

2.2.8 Wide & Deep

Wide & Deep is a technique that uses a wide neural network to learn the preferences of the customer and utilizes another deep neural network to learn the products's features. The wide model is generalized linear model (GLM) and the deep model is a dense neural network (DNN).

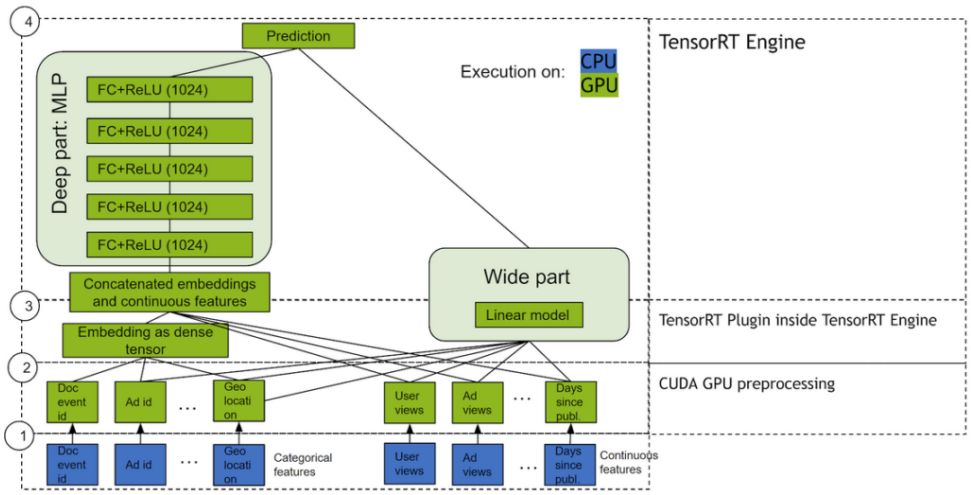


Figure 2.5: Wide Deep Structure
[6]

2.2.9 DLRM

Deep Learning Recommendation Model (DLRM) it is a technique that uses a deep neural network to handle categorical and numerical features. each categorical feature is represented as a one-hot vector and each numerical feature is represented as a dense vector, both fed into multi-layer perceptron (MLP) layers. The output of the MLP layers is then fed into a dot product layer to compute the inner product of the feature vectors. The output of the dot product layer is then fed into a sigmoid layer to compute the probability of the user liking the item.

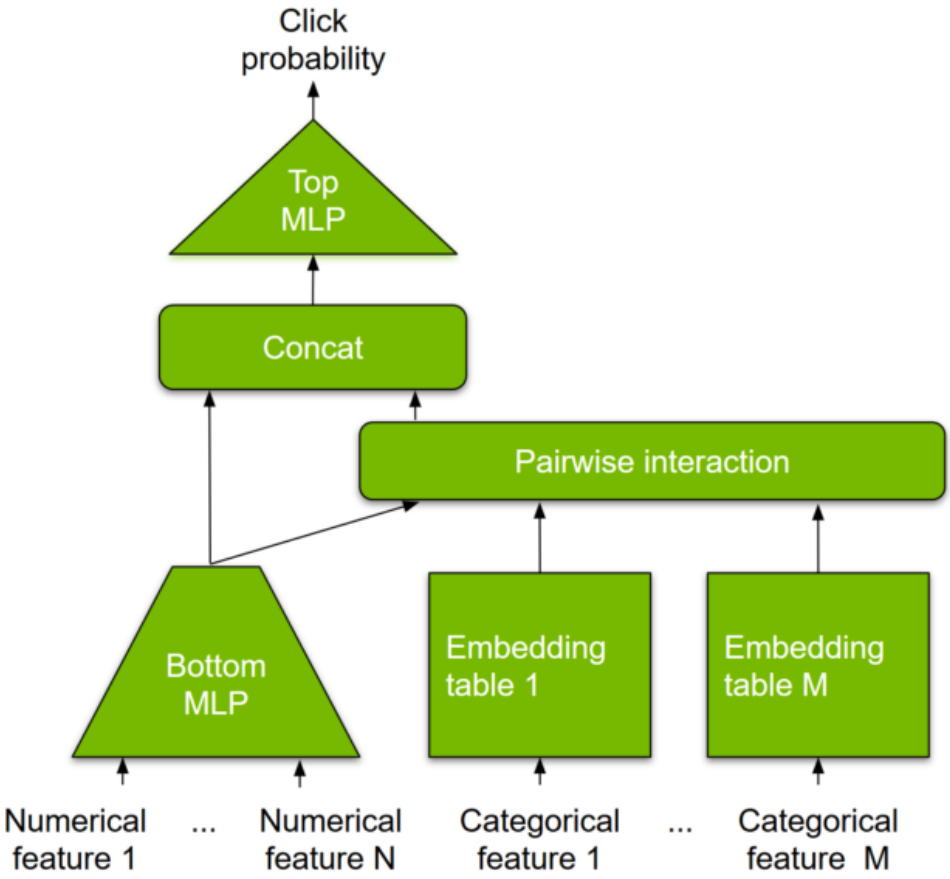


Figure 2.6: DLRM Structure [6]

Chapter 3

Requirements & Literature Review

Contents

3.1 Functional Requirements

The system should provide a RESTful API as the final interface to be used by the front-end application. The API provides endpoints that allow inserting customers, products, and interactions. In addition to endpoints for retrieving the recommendations for a given customer.

3.2 System Requirements

In order to for the system to be useful it has to meet the following specifications:

3.2.1 Scalability

Scalability implies that it has to be cloud-native, the inference system should apply proper load balancing across multi-node, multi model deployments.

3.2.2 Real-time predictions

To be usable in any website or application, the system should be able to provide real-time predictions, suggestions, with few milliseconds latency.

To fulfil this requirement, trained models should run on optimized inference servers or services, the suggested deployment plan is to use **Nvidia Triton** [1] inference server [10], integrated with **Amazon SageMaker** model deployment[11] as infrastructure.

3.2.3 Near Real-time Training

This implies continuous training and deployment of model which requires the automation of training and deployment.

3.2.4 Elasticity & Optimization

Elasticity is vital for keeping up with traffic spikes and declines while optimizing infrastructure costs. To achieve this, the system should be able to scale up and down based on the traffic and load.

3.2.5 Security

Like any other system, the system has to be immune to security threats by implementing best practices at every level in the deployment and design.

e.g rate-limiting requests to interaction injection endpoints, using attestation when possible, limiting access to user and product CRUD operations.

3.3 Related Work

There are many open-source and paid solutions that provide recommendation systems and libraries, this section discusses some of them.

3.3.1 LightFM [1]

A Python library that enables the classic matrix factorization techniques to include metadata about both items and users, incorporating both content and collaborative information into the recommendation process (hybrid).

Its approach is described with more depth in the LightFM paper [12].

[1]Nvidia Triton Inference Server, part of the Nvidia AI platform and available with Nvidia AI Enterprise, is open-source software that standardizes AI model deployment and execution across every workload.

3.3.2 Rexy [2]

Rexy is a Python library that provides a general-purpose recommendation system framework. It is flexible and can be adapted to a variety of data schemas.

3.3.3 Gorse [2]

Gorse is an open source recommender system engine implemented in Go that provides a scalable and flexible recommendation system framework. It supports a variety of algorithms, including collaborative filtering, content-based filtering, and deep learning.

3.3.4 AWS Personalize [3]

"Amazon Personalize allows developers to quickly build and deploy curated recommendations and intelligent user segmentation at scale using machine learning (ML). Because Amazon Personalize can be tailored to your individual needs, you can deliver the right customer experience at the right time and in the right place." [2]

3.3.5 Google Recommendations AI [4]

Google describes it as "Recommendations AI enables you to build an end-to-end personalized recommendation system based on state-of-the-art deep learning ML models, without a need for expertise in ML or recommendation systems."[3]

3.3.6 Nvidia Merlin [5]

"NVIDIA Merlin is an open source library providing end-to-end GPU-accelerated recommender systems, from feature engineering and preprocessing to training deep learning models and running inference in production." [4]

The frameworks, discussed in more depth later, provides many components including:

- Merlin Models [15]
- Merlin NVTabular [16]
- Merlin HugeCTR [17]
- Merlin Transformers4Rec [18]
- Merlin SOK (SparseOperationsKit)
- Merlin Distributed Embeddings (DE)
- Merlin Systems [19]

Making it a very customizable and extensible solution.

[2]AWS description of the service [3]
[3]Google Cloud Marketplace [13]
[4]Nvidia Merlin Repository [14]

Table 3.1: Comparison of Recommendation Solutions

System	**LightFM**
License	Apache 2.0
Algorithm Type	Matrix Factorization
Hardware Utilization	CPU
Deployment Readiness	Library (Additional Components Needed)
Notes	-
System	**Rexy**
License	MIT
Algorithm Type	Matrix Factorization
Hardware Utilization	CPU
Deployment Readiness	Library (Additional Components Needed)
Notes	-
System	**Gorse**
License	Apache 2.0
Algorithm Type	Matrix Factorization
Hardware Utilization	CPU
Deployment Readiness	Single-node-learning multi-node-inference cluster
Notes	Unreliable and has many bugs
System	**AWS Personalize**
License	Proprietary
Algorithm Type	DLRM
Hardware Utilization	-
Deployment Readiness	A lot of customization required
Notes	High customization, predictions, and training fees
System	**Google Recommendations AI**
License	Proprietary
Algorithm Type	DLRM
Hardware Utilization	-
Deployment Readiness	End-to-End service
Notes	High predictions, and training fees
System	**Nvidia Merlin**
License	Apache 2.0
Algorithm Type	Multiple Options
Hardware Utilization	Optimized for Nvidia GPUs
Deployment Readiness	Recommendation pipelines components
Notes	Very customizable

Bibliography

[1] "LightFM." https://making.lyst.com/lightfm/docs/home.html, accessed: 2023-10-8.

[2] "Rexy." https://github.com/kasraavand/Rexy, accessed: 2023-10-8.

[3] AWS, "AWS Personalize." https://aws.amazon.com/personalize/, accessed: 2023-10-4.

[4] Google Cloud, "Google Recommendations AI." https://cloud.google.com/recommendations. Accessed: 2023-11-9.

[5] Nvidia, "Merlin." https://developer.nvidia.com/merlin, accessed: 2023-10-6.

[6] Nvidia Glossary, "Recommendation System." https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/. Accessed: 2023-12-19.

[7] Salesforce Marketing Cloud, "Predictive intelligence benchmark report." https://brandcdn.exacttarget.com/sites/exacttarget/files/deliverables/etmc-predictiveintelligencebenchmarkreport.pdf, 2014.

[8] Google Cloud Summit, "Ikea's approach to building a powerful recommendations engine." by Google Cloud Events https://www.youtube.com/watch?v=PyjC0wRRtBg, 2021.

[9] Sungoh Park and Kyoungtae Hwang, "Increasing customer engagement and loyalty with personalized coupon recommendations using Amazon Personalize," *AWS Machine Learning Blog*, 2020.

[10] Nvidia, "Nvidia Triton Inference Server." https://developer.nvidia.com/triton-inference-server, accessed: 2023-10-4.

[11] AWS, "Amazon SageMaker Model Deployment." https://aws.amazon.com/sagemaker/deploy/, accessed: 2023-10-4.

[12] M. Kula, "Metadata embeddings for user and item cold-start recommendations," 2015.

[13] "Google Cloud Marketplace - Recommendation AI." https://console.cloud.google.com/marketplace/product/google/recommendationengine.googleapis.com, accessed: 2023-12-20.

[14] Nvidia, "Merlin Repository." https://github.com/NVIDIA-Merlin/Merlin, accessed: 2023-10-6.

[15] Nvidia, "Merlin Models Repository." https://github.com/NVIDIA-Merlin/models, accessed: 2023-10-6.

[16] Nvidia, "Merlin NVTabular." https://developer.nvidia.com/nvidia-merlin/nvtabular, accessed: 2023-10-6.

[17] Nvidia, "Merlin HugeCTR." https://developer.nvidia.com/nvidia-merlin/hugectr, accessed: 2023-10-6.

[18] Nvidia, "Merlin Transformers4Rec." https://github.com/NVIDIA-Merlin/Transformers4Rec, accessed: 2023-10-6.

[19] Nvidia, "Merlin Systems Repository." https://github.com/NVIDIA-Merlin/systems, accessed: 2023-10-6.