# RecPress: Knowledge Distillation for Efficient Recommender Systems

Deepak
MA20BTECH11019
ma20btech11019@iith.ac.in

Tapishi Kaur
MA20BTECH11017
ma20btech11017@iith.ac.in

Safder Shakil
EM23MTECH11007
em23mtech11007@iith.ac.in

Abdul Waris
CE22RESCH11001
ce22resch11001@iith.ac.in

## 1. Motivation

Recommender systems have become an essential part of many online platforms, such as Netflix, Amazon, Youtube. The increasing importance of recommender systems has led to a growing need for efficient recommender systems. Efficient not only in terms of the accuracy but also in terms of scalability, accessibility and other factors. This can be seen clearly from the celebrated Netflix prize, where the prize-winning model (ensemble of around 160 models) despite its remarkable accuracy, remained too large and intricate for practical deployment nevertheless the event has made great contributions to the development in field of recommender systems.

Meanwhile the rise of edge computing has led to active research on tiny-ML, which aims to build small but efficient machine learning systems, one of such many interesting methods is Knowledge distillation.

Our project aims to tackle this dilemma of balancing accuracy and other factors for efficiency by harnessing the concept of knowledge distillation. We endeavor to distill the extensive wisdom embedded within a large, intricate teacher model, such as the state-of-the-art recommender system, into a sleeker and faster student model.

## 2. Problem Statement

The project aims to apply knowledge distillation techniques to develop efficient recommender systems and bridge the gap between high-accuracy but resource-intensive recommender systems and practical, efficient solutions that may also be deployed on edge devices and similar resource-constrained environments while maintaining the quality of recommendations.

## 3. Objectives

The current objectives and tasks of the project are as follows

**Knowledge Extraction** Analyze and understand the decision-making processes and feature extraction mechanisms of the KD teacher model. Understand and implement existing methods to extract relevant knowledge, patterns, and recommendations from the teacher model's predictions and internal representations.

**Student Model Development** Understanding the Design and implement a compact student recommender system that can produce high-quality recommendations. Train the student model using the extracted knowledge from the teacher model.

**Evaluation** Assess the recommendation quality of the student model in terms of accuracy and user satisfaction compared to the teacher model. Measure the efficiency gains achieved by the student model in terms of scalability, reduced model size, computational resources, and faster inference times.

## 4. Literature Survey

### 4.1. Deep Recommender Systems

Recommender systems learn about people's likes and dislikes, past choices, and other features by looking at how they interact with products, such as by viewing them, clicking on them, liking them, or buying them.

These systems can be classified into three main categories based on the primary techniques and data sources they use to make recommendations.

1. **Collaborative Filtering:** Collaborative filtering methods are founded on the idea that users who have shown similar behaviors or preferences in the past will likely have similar preferences in the future. This approach relies on user-item interactions to make recommendations. There are two primary types of collaborative filtering: user-based and item-based.

2. **Content-Based Filtering:** Content-based filtering recommends items to users based on the attributes of the items and a profile of the user's preferences. This approach relies on extracting features or keywords from items and matching them to a user profile.

3. **Hybrid Recommender Systems:** Hybrid systems combine collaborative and content-based filtering methods to leverage the strengths of both. These systems aim to improve recommendation accuracy and overcome some of the limitations of individual approaches

Recommender systems are employed to address a wide array of specific problems and challenges in various domains, such as Click-Through Rate (CTR) Prediction, Next-Item Recommendation, Item Ranking, Cold-Start Problem, Session-Based Recommendations, Multi-Armed Bandit Problems, Cross-Domain Recommendations etc...

In the context of our project we are considering the papers majorly trying to solve two main problems

1. Next-Item Recommendation

2. Click-Through Rate (CTR) Prediction

3. Top N recommendation

#### 4.1.1 Papers

**1. Neural Collaborative Filtering** The introduction of Neural Collaborative Filtering (NCF) by He et al. in 2017 marked a significant breakthrough in the field of recommender systems, steering the course of recommendation technology towards the realm of deep learning. NCF emerged as a transformational departure from the conventional matrix factorization approach, revolutionizing the way we approach user-item interactions and recommendation accuracy.

Before NCF, the prevailing standard in recommender systems was matrix factorization. In this framework, latent vectors or embeddings were learned for both users and items, and recommendations were made by computing the dot product between the user vector and item vectors. The closer the dot product approached 1, the better the match. In essence, matrix factorization was akin to a linear model of latent factors, making it a solid foundation but inherently limited by its linear assumptions.

NCF's ingenious innovation was to replace the linear inner product in matrix factorization with a neural network. This transformation involved concatenating the embeddings of users and items, followed by passing them through a multi-layer perceptron (MLP) with a single task head, which predicted user engagement, such as clicks. During model training, both the weights of the MLP and the embedding weights, responsible for mapping IDs to their respective embeddings, were learned through backpropagation of loss gradients.

The fundamental premise underpinning NCF was a departure from the linear assumption of user-item interactions in matrix factorization to embrace non-linearity. The hypothesis posited that user-item interactions are inherently non-linear, and by introducing additional layers to the MLP, performance gains could be achieved. Indeed, the results validated this notion. He et al. demonstrated that with just four layers, NCF outperformed the best matrix factorization algorithms of its time by a substantial margin, achieving a remarkable 5% improvement in hit rate on benchmark datasets like Movielens and Pinterest.

**2. Wide and Deep Learning for Recommender Systems** The "Wide & Deep Learning for Recommender Systems" paper, published by Google in 2016, introduced an innovative approach to enhance recommendation systems. It tackled a critical missing element in earlier models: cross features.

Cross features are essentially combinations of two original features. What makes them powerful is their ability to capture relationships between different levels of features. This means they can remember fine-grained details (memorization) and also generalize well (generalization).

The core idea of Wide & Deep Learning is to blend two key components, **Wide Module** which focuses on memorization, it uses a linear layer to directly take all cross features as inputs which helps the model remember specific user-item interactions and **Deep Module**, it is like the one used in Neural Collaborative Filtering (NCF). It uses deep neural networks to capture complex, non-linear patterns and relationships. What's remarkable about this approach is that it combines both modules into a single output task head. By doing so, it creates a well-rounded model that can remember intricate details while also understanding broader trends in user engagement

**3. DeepFM** "DeepFM"(2017) is Huawei's innovative approach to enhancing recommendation systems, offering a unique twist on the traditional architecture. This method presents a notable departure from earlier models and introduces a dedicated neural network to learn cross features in the wide component.

In "DeepFM," the wide component is not a cross neural network, as seen in some previous architectures, but rather a "Factorization Machine" (FM) layer which computes the dot-products of all pairs of embeddings.

The model learns embeddings for each of the features for user and different types of product attributes, and the FM layer comes into play by calculating dot products corresponding to various combinations of user-attribute,

attribute-attribute. Essentially, it revisits the concept of matrix factorization in a new light.

The output of the FM layer is then seamlessly combined with the output of the deep component. This fusion leads to an output that is activated using a sigmoid function, resulting in the model's prediction.

**4. BERT4Rec** introduced in the paper "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformers" by Sun et al. (2019), marks a significant stride in recommender systems. This model leverages the power of BERT, a cutting-edge natural language processing model, to enhance sequential recommendation. BERT4Rec's core concept involves representing users and items as sequences of tokens, be it words or product IDs. By employing BERT, it derives bidirectional representations for these sequences, capturing both token context and relationships. With these bidirectional representations in hand, BERT4Rec computes a similarity score between the user and each item, signifying the user's potential interest. Recommendations are then made based on these scores. What sets BERT4Rec apart are its distinct advantages. It adopts bidirectional representations, enabling a more profound understanding by considering both token context and relationships. It's adept at understanding item relationships across various domains, thanks to its grasp of the semantic meanings within item sequences. BERT4Rec represents a significant leap in recommender systems, consistently outperforming state-of-the-art models. Its versatility spans multiple domains, and its ease of training and deployment makes it a promising tool.

## 4.2. Knowledge Distillation

Literature survey from the following papers:

- Distilling the Knowledge in a Neural Network (2015) Hinton et al.
- Knowledge Distillation: A Survey (2021) Jianping Gou et al.

**Knowledge Distillation** involves the transfer of knowledge from one or more large, complex models to a single, more compact model that can be efficiently deployed within practical real-world limitations. A knowledge distillation setup comprises three fundamental elements: the knowledge itself, the distillation algorithm, and the teacher-student architecture.

### 4.2.1 Types of Knowledge

**Response-based knowledge** is a type of knowledge focuses on the actual outputs or responses produced by a model for a given input, it often involves the probability distributions or class predictions generated by a larger, teacher model with a goal to transfer this output distribution to a smaller student model, enabling it to make similar predictions.

**Feature-based knowledge** emphasizes the intermediate representations or features learned by a model during the course of its training. Instead of directly transferring output responses, this approach aligns the internal representations of the teacher and student models. By doing so, the student model can capture meaningful features from the data, improving its overall performance.

**Relation-based knowledge** pertains to the relationships and dependencies between various elements within the data or model. It involves understanding how different parts of the input are interconnected or how certain features are related to each other. Knowledge distillation in this context may aim to transfer knowledge about these relationships, helping the student model make more informed decisions.

### 4.2.2 Various Distillation Algorithms

The papers contains different kinds of distillation algorithms, of which following the ones we would try to incorporate into project.

1. **Neural architecture search-based distillation** uses NAS algorithm to find an efficient student model architecture then perform KD.
2. **Adversarial Distillation** introduces adversarial training into the distillation process. The student model learns to generate outputs that not only match the teacher's predictions but also resist adversarial attacks, enhancing robustness.
3. **Attention-based distillation** transfers knowledge about where the teacher model focuses its attention during inference. The student model learns to attend to similar regions in the input data, aiding its understanding.
4. **Graph-based distillation**, represents knowledge using graph structures, such as knowledge graphs or semantic graphs. This enables the transfer of structured knowledge and relationships between concepts.
5. **Multi-teacher distillation** involves using multiple teacher models to provide diverse knowledge. The student model learns from the collective knowledge of these teachers, potentially improving performance.

#### 4.2.3 Types of training methods

1. **Offline Distillation:** The teacher model is trained on a large dataset, producing soft targets or other knowledge representations. Once the teacher is trained, the student model is trained on the same dataset using the knowledge distilled from the teacher.

2. **Online Distillation:** Both the teacher and student models are updated iteratively as the training data is processed. The student model aims to mimic the teacher's outputs or knowledge representations as the training progresses. This approach can be computationally efficient and adaptable to changing data distributions.

3. **Self Distillation** has the same teacher and student architecture, the model train itself with different aspects from the previous trainings such as deeper layers of a neural network can be used to train the shallow layers etc..

### 4.3. Knowledge Distillation for RecSys

Literature survey of papers on Knowledge distillation of various recommender systems.

- **DE-RRD: A Knowledge Distillation Framework for Recommender System**

  The research paper "DE-RRD: A Knowledge Distillation Framework for Recommender Systems" authored by SeongKu Kang, Junyoung Hwang, and Hwanjo Yu introduces an innovative knowledge distillation framework tailored for recommender systems, known as DE-RRD, it not only allows the student model to glean insights from the teacher model's latent knowledge but also benefits from the teacher's predictive capabilities.

  DE-RRD encompasses two distinctive techniques:

  **Distillation Experts (DE):** The DE method facilitates the direct transfer of latent knowledge from the teacher model to the student model. It leverages the concept of "experts" and employs a novel expert selection strategy, ensuring the effective distillation of the extensive knowledge held by the teacher model, even within the constraints of the student model's capacity.

  **Recommendation Distillation (RD):** RD, on the other hand, focuses on distilling the teacher's predictions into the student model. This is achieved through the utilization of a unique loss function that takes into account the uncertainty associated with the teacher's predictions.

  DE-RRD's efficacy is thoroughly assessed using three publicly available datasets, with a specific emphasis on top-N recommendation tasks. The results clearly demonstrate that DE-RRD surpasses several state-of-the-art knowledge distillation methods designed for recommender systems in terms of essential metrics such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG).

  We would continue our literature towards some more frameworks combining the Recommender systems with knowledge distillation as follows and Try to get insights on how they vary based on the problem the recommender system is trying to solve and the architecture of recommender system.

  **Following are the papers we are going to study further on:**

- Scene-adaptive Knowledge Distillation for Sequential Recommendation via Differentiable Architecture Search.
- On-Device Next-Item Recommendation with Self-Supervised Knowledge Distillation
- A novel Enhanced Collaborative Autoencoder with knowledge distillation for top-N recommender systems.

## 5. References


1. Neural Collaborative Filtering (2017) Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua
2. Wide & Deep Learning for Recommender Systems (2016) Heng-Tze Cheng et al.
3. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction (2017) Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He
4. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer (2019) Fei Sun et al.
5. Distilling the Knowledge in a Neural Network (2015) Geoffrey Hinton et al
6. Knowledge Distillation: A Survey (2021) Jianping Gou et al.
7. DE-RRD: A Knowledge Distillation Framework for Recommender System SeongKu Kang et al (2020).
8. A novel Enhanced Collaborative Autoencoder with knowledge distillation for top-N recommender systems. Pan, Yiteng et al. Neurocomputing 2019
9. Scene-adaptive Knowledge Distillation for Sequential Recommendation via Differentiable Architecture Search (2021) Lei chen et al.