



# DEEP-LEARNING BASED RECOMMENDER SYSTEMS

Giảng viên hướng dẫn: Ngô Minh Mẫn

Võ Thái Bình - 20280007

Hoàng Hải Đăng - 20280011

# Nội dung thuyết trình

## Phần 1

*Giới thiệu*

## Phần 4

*Mô Hình Đề Xuất  
Tuần Tự  
(Sequential  
Recommendation)*

## Phần 2

*Tổng quan về  
Recommender  
System và Deep  
Learning*

## Phần 5

*Thực Nghiệm*

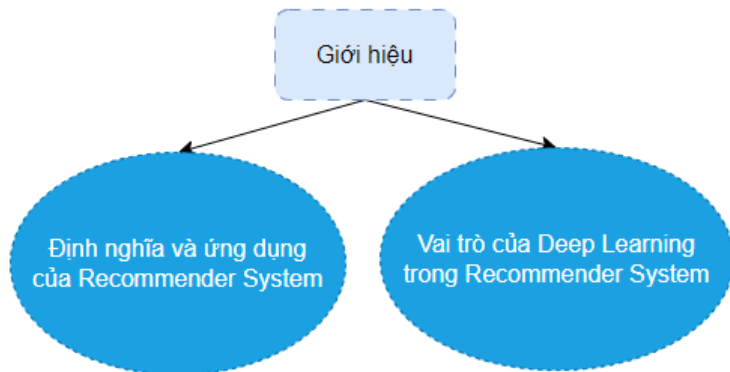
## Phần 3

*Thách Thức Và  
Cơ Hội*

## Phần 6

*Kết Luận Và  
Hướng Đi Trong  
Tương Lai*

# Phần 1: Giới thiệu:



# Phần 2: Tổng quan về Recommender System và Deep Learning

## Phần 2.1

*Các kĩ thuật đề xuất truyền thống trong Recommender System*

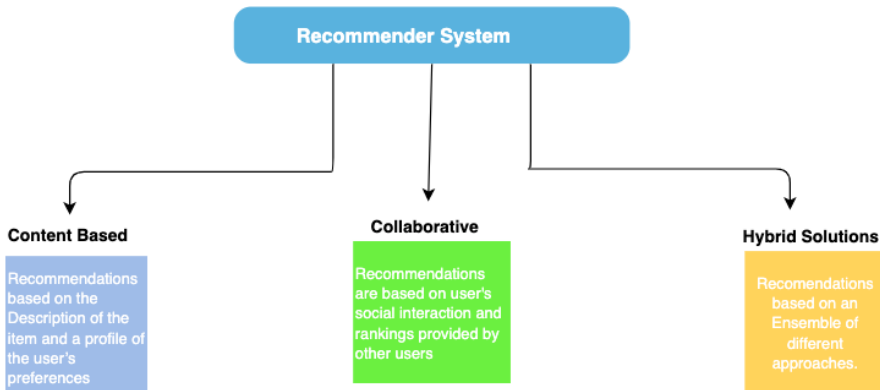
## Phần 2.2

*Deep-learning based Recommender System*

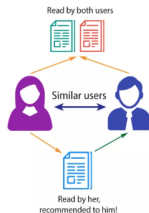
## Phần 2.3

*Quá trình phát triển của Deep-learning based Recommender System*

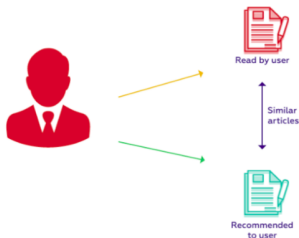
## 2.1 Các kĩ thuật đề xuất truyền thống trong Recommender System



# Collaborative Filtering

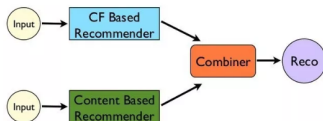


- **Định nghĩa:** Collaborative Filtering là một phương pháp đề xuất dựa trên hành vi của cả nhóm user. Nó giả định rằng user có sở thích tương tự sẽ thích những item tương tự.
- **Loại hình:** Có hai dạng chính: User-based Collaborative Filtering (dựa trên sự giống nhau giữa các user) và Item-based Collaborative Filtering (dựa trên sự giống nhau giữa các item).
- **Ưu điểm:** Đơn giản triển khai và không đòi hỏi thông tin chi tiết về user và item.



- **Định nghĩa:** Content-based Filtering tập trung vào đặc tính của các item và sở thích cá nhân của user. Nó đề xuất các item dựa trên mức độ tương quan giữa nội dung của item và sở thích của user.
- **Loại hình:** Xác định các đặc tính của item và tạo ra mô hình dựa trên sở thích của user với các đặc tính đó.
- **Ưu điểm:** Hoạt động tốt trong việc giải quyết vấn đề dữ liệu thưa và có khả năng giúp giải quyết cold start problem cho các item mới.

# Hybrid Method



- **Định nghĩa:** Hybrid Method kết hợp Collaborative Filtering và Content-based Filtering để tận dụng ưu điểm của cả hai phương pháp.
- **Loại hình:** Có thể được chia thành hai loại chính: Model Fusion (kết hợp dự đoán từ các mô hình riêng biệt) và Feature Combination (kết hợp đặc tính từ cả hai loại phương pháp).
- **Ưu điểm:** Có thể giảm đi nhược điểm của từng phương pháp đơn lẻ và cung cấp độ chính xác cao hơn.



# Giới hạn

- **Dữ Liệu Thưa (Data Sparsity):** Collaborative Filtering gặp khó khăn khi dữ liệu thưa, trong khi Content-based Filtering có thể không có đủ thông tin để tạo ra đề xuất chính xác.
- **Cold Start Problem:** Cả hai phương pháp đều gặp vấn đề khi phải đối mặt với user mới hoặc item mới mà chưa có đánh giá nào.
- **Over-Specialization:** Content-based Filtering có thể tạo ra đề xuất quá chuyên sâu và hạn chế đa dạng của các item.
- **Challenges in Combining Methods:** Hybrid Method đôi khi đòi hỏi các thuật toán phức tạp và không phải lúc nào cũng đem lại cải tiến đáng kể so với các phương pháp đơn lẻ.

## 2.2 Deep-learning based Recommender System

### Định nghĩa

*Deep-learning based Recommender System là những hệ thống đề xuất mà sử dụng các mô hình học sâu (Deep Learning) để hiểu và mô hình hóa mối quan hệ phức tạp giữa user và các item. Trong Deep-learning based Recommender System, các biểu diễn ẩn phức tạp của user và item được học từ dữ liệu để tạo ra các dự đoán chính xác về sự tương tác hoặc sự quan tâm của user đối với các item chưa được xem.*

## Một số mô hình Deep-learning được sử dụng phổ biến trong Recommender System:

- Artificial Neural Networks (ANN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- BERT (Bidirectional Encoder Representations from Transformers)
- Wide & Deep Learning
- Matrix Factorization with Neural Networks (MFNN)

## 2.3 Quá trình phát triển của Deep-learning based Recommender System

Deep Learning qua thời gian đã đem lại nhiều đổi mới và cải tiến đáng kể trong lĩnh vực Recommender System. Dưới đây là một số cột mốc quan trọng trong quá trình phát triển của Deep Learning trong hệ thống đề xuất:

- **2014 - Collaborative Deep Learning for Recommender Systems**
- **2016 - Wide & Deep Learning for Recommender Systems**
- **2018 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
- **2020 - Neural Collaborative Filtering vs. Matrix Factorization Revisited**

## Phần 3: Thách Thức Và Cơ Hội

### Thách thức:

- **Data Sparsity:** Gặp khó khăn do số lượng đánh giá của user hạn chế, dẫn đến dữ liệu thưa.
- **Cold Start Problem:** Cần một lượng lớn dữ liệu mới để có thể bắt đầu hoạt động hiệu quả sau một khoảng thời gian không hoạt động hoặc khi mới triển khai.
- **Explainability:** Các mô hình Deep Learning thường khó hiểu, giảm khả năng giải thích tại sao một đề xuất được thực hiện.
- **Scalability:** Một số mô hình Deep Learning đòi hỏi nguồn lực tính toán đáng kể, tạo ra vấn đề về khả năng mở rộng với số lượng lớn user và item.

# Thách Thức Và Cơ Hội

## Cơ hội:

- **Đa dạng và Cá nhân hóa:** Deep Learning mang lại cơ hội tạo ra các mô hình phức tạp có khả năng đề xuất các item phù hợp với sở thích đa dạng của user.
- **Ứng dụng đa lĩnh vực:** Deep Learning có thể được áp dụng không chỉ trong giải trí mà còn trong thương mại điện tử, giáo dục, chăm sóc sức khỏe và nhiều lĩnh vực khác.
- **Khả năng thích ứng:** Deep Learning có thể được thiết kế để học liên tục, giúp chúng thích ứng với sự biến đổi của sở thích user theo thời gian.
- **Độ trễ thấp:** Các kiến trúc và tối ưu hóa tiên tiến của Deep Learning cho phép hệ thống đề xuất theo thời gian thực.

## Phần 4: Mô Hình Đề Xuất Tuần Tự (Sequential Recommendation)

### Mô hình đề xuất tuần tự

Là một hệ thống đề xuất được thiết kế để xử lý dữ liệu tuần tự, nơi mà thông tin thời gian có ý nghĩa quan trọng. Trong ngữ cảnh này, dữ liệu thường được tổ chức theo dạng chuỗi thời gian, chẳng hạn như lịch sử tương tác của user với sản phẩm hoặc nội dung trong một khoảng thời gian dài.

### Một số mô hình đề xuất tuần tự

- BERT4Rec (BERT for Recommender Systems)
- SASRec (Self-Attentive Sequential Recommendation)
- Caser (Convolutional Sequence Embedding Recommendation Model)

## 4.1 BERT4Rec (BERT for Recommender Systems)

BERT4Rec sử dụng mô hình ngôn ngữ BERT để cải thiện khả năng hiểu nội dung và ý định user trong hệ thống đề xuất. BERT4Rec giải quyết vấn đề sequential recommendation và có thể áp dụng cho nhiều ngữ cảnh.



# Network Architecture

# Transformer Layer

# Multi-Head Self-Attention

# Position-wise Feed-Forward Network

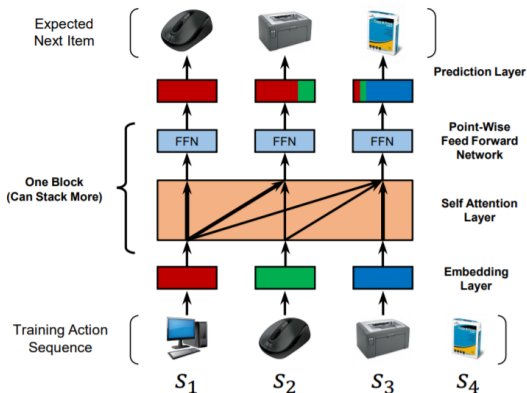
# Stacking Transformer Layer

# Embedding Layer

# Model Learning

## 4.2 SASRec (Self-Attentive Sequential Recommendation)

SASRec sử dụng cơ chế self-attention (self-attention) để hiệu quả hóa quá trình đề xuất theo chuỗi thời gian.





# Embedding Layer

# Positional Embedding

# Self-Attention Block

# Self-Attention layer

# Network Training

## 4.3 Caser (Convolutional Sequence Embedding Recommendation Model)

Caser sử dụng các lớp convolutional để học các đặc trưng từ chuỗi thời gian của sự kiện. Điều này giúp mô hình hiểu được các mối quan hệ không chỉ giữa các sự kiện liên tiếp mà còn giữa các sự kiện không liên tiếp trong chuỗi.

# Network Training

# Phần 5: Thực Nghiệm

## Phần 5.1

Mô tả bài toán

## Phần 5.2

Benchmark Dataset

## Phần 5.3

Input & Output

## Phần 5.4

Task Settings & Evaluation Metrics

## Phần 5.5

Triển Khai Chi Tiết

## Phần 5.6

So Sánh Hiệu Suất Tổng Thể

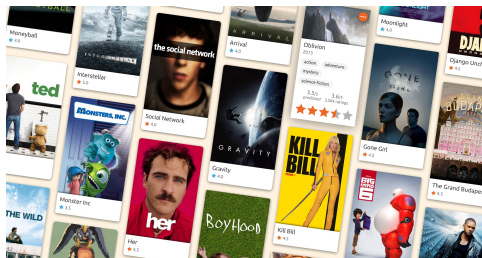


## 5.1 Mô tả bài toán

Ta đặt  $\mathcal{U} = u_1, u_2, \dots, u_{|\mathcal{U}|}$  là tập các user,  $\mathcal{V} = v_1, v_2, \dots, v_{|\mathcal{V}|}$  là tập các item và danh sách  $\mathcal{S}_u = [v_1^{(u)}, \dots, v_t^{(u)}, \dots, v_{n_u}^{(u)}]$  là chuỗi tương tác theo thứ tự thời gian của user  $u \in \mathcal{U}$ , trong đó  $v_t^{(u)} \in \mathcal{V}$  là item mà  $u$  đã tương tác tại time step  $t$  và  $n_u$  là độ dài của chuỗi tương tác của user  $u$ . Bài toán đặt ra là cho lịch sử tương tác  $\mathcal{S}_u$ , hệ thống đề xuất tuần tự sẽ dự đoán item mà user  $u$  sẽ tương tác tại time step  $n_u + 1$ . Bài toán có thể được công thức hoá như sau:

$$p(v_{n_u+1}^{(u)} = v | \mathcal{S}_u)$$

## 5.2 Benchmark Dataset



Bộ dữ liệu MovieLens là một bộ dữ liệu nổi tiếng trong lĩnh vực hệ thống đề xuất và nghiên cứu lọc cộng tác. Nó thường được sử dụng để kiểm thử và đánh giá hiệu suất của các thuật toán đề xuất khác nhau. Bộ dữ liệu này bao gồm các đánh giá phim được user thực hiện, cùng với thông tin bổ sung về các bộ phim và user.

Có nhiều phiên bản của bộ dữ liệu MovieLens với kích thước và chi tiết khác nhau. Trong bài này, chúng ta sẽ tập trung vào hai phiên bản:

MovieLens 1m (**ML-1m**) and MovieLens 20m (**ML-20m**)

## 5.3 Input & Output

Tất cả đánh giá được chứa trong file `rating.csv`, mỗi dòng đại diện cho một đánh giá của một bộ phim bởi một người dùng và có định dạng sau:

```
userId,movieId,rating,timestamp
```

Do đó, input cho mô hình là một chuỗi tương tác của user bao gồm các item và timestamp tương ứng; output của mô hình sẽ là item có xác suất cao nhất mà người dùng sẽ tương tác ở time step tiếp theo.

## 5.4 Task Settings & Evaluation Metrics

### Để đánh giá hiệu suất của tất cả các mô hình

Chúng tôi sử dụng nhiều thước đo đánh giá, bao gồm Hit Ratio (HR), Normalized Discounted Cumulative Gain (NDCG), và Mean Reciprocal Rank (MRR). Xem xét việc chúng ta chỉ có một item thực tế cho mỗi user,  $HR@k$  tương đương với  $Recall@k$  và tỷ lệ thuận với  $Precision@k$ ; MRR tương đương với Mean Average Precision (MAP). Trong bài này, chúng tôi trình bày HR và NDCG với  $k = 1, 5, 10$ . Đối với tất cả các thước đo này, giá trị càng cao thì hiệu suất càng tốt.

## 5.5 Triển Khai Chi Tiết

### Đối với SASRec<sup>a</sup> và Caser<sup>b</sup>

<sup>a</sup><https://github.com/kang205/SASRec>

<sup>b</sup>[https://github.com/graytowne/caser\\_pytorch](https://github.com/graytowne/caser_pytorch)

Chúng tôi sử dụng code được cung cấp bởi tác giả tương ứng. Đối với các siêu tham số chung trong tất cả các mô hình, chúng tôi xem xét kích thước chiều ẩn  $d$  từ  $\{16, 32, 64, 128, 256\}$ , hệ số điều chỉnh  $\ell_2$  từ  $\{1, 0.1, 0.01, 0.001, 0.0001\}$  và tỷ lệ dropout từ  $\{0, 0.1, 0.2, \dots, 0.9\}$ . Tất cả các siêu tham số khác (ví dụ như thứ tự Markov trong Caser) và các phương pháp khởi tạo đều tuân theo gợi ý từ các tác giả hoặc được điều chỉnh trên các tập validation. Chúng tôi thể hiện kết quả của mỗi mô hình dưới bộ siêu tham số tối ưu của nó.

## 5.5 Triển Khai Chi Tiết

### Đối với BERT4Rec<sup>a</sup>

<sup>a</sup><https://github.com/FeiSun/BERT4Rec>

Chúng tôi thực hiện với TensorFlow. Tất cả tham số được khởi tạo bằng truncated normal distribution trong khoảng  $[-0.02, 0.02]$ . Chúng tôi train mô hình dùng Adam optimizer với learning rate  $1e - 4$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\ell_2$  weight decay 0.01, và linear decay của learning rate. Đạo hàm bị cắt khi  $\ell_2$  norm đạt ngưỡng 5.

## 5.5 Triển Khai Chi Tiết

### Để so sánh công bằng

Chúng tôi cho số layer  $L = 2$ , head number  $h = 2$  và chiều dài tối đa của chuỗi hành vi  $N = 200$ . Đối với cài đặt head, chúng tôi cho thực nghiệm số chiều của mỗi head là 32 (single head nếu  $d < 32$ ). Chúng tôi điều chỉnh mask proportion  $\rho$  dùng tập validation, kết quả là  $\rho = 0.2$  cho ML-1m và ML-20m. Tất cả mô hình đều được train từ đầu trên một GPU NVIDIA GeForce GTX 1080 Ti với batch size là 256.

## 6.5 So Sánh Hiệu Suất Tổng Thể

Bảng bên dưới tóm tắt kết quả tốt nhất của tất cả các mô hình trên hai bộ dữ liệu thử nghiệm. Chúng tôi bỏ qua kết quả nDCG@1 vì nó bằng với HR@1 trong các thực nghiệm.

Datasets	Metric	Caser	SASRec	BERT4Rec
ML-1m	HR@1	0.2194	0.2351	0.2863
	HR@5	0.5353	0.5434	0.5876
	HR@10	0.6692	0.6629	0.6970
	NDCG@5	0.3832	0.3980	0.4454
	NDCG@10	0.4268	0.4368	0.4818
	MRR	0.3648	0.3790	0.4254
ML-20m	HR@1	0.1232	0.2544	0.3440
	HR@5	0.3804	0.5727	0.6323
	HR@10	0.5427	0.7136	0.7473
	NDCG@5	0.2538	0.4208	0.4967
	NDCG@10	0.3062	0.4665	0.5340
	MRR	0.2529	0.4026	0.4785

Theo kết quả, rõ ràng là BERT4Rec thể hiện tốt nhất trong 3 phương pháp trên hai bộ dữ liệu thử nghiệm đối với tất cả các evaluation metric.



## Phần 6: Kết Luận Và Hướng Đi Trong Tương Lai

### Tổng kết

*Việc khám phá về hệ thống đề xuất dựa trên Deep Learning đã mở ra những cơ hội hứa hẹn để tạo ra các hệ thống đề xuất chính xác và cá nhân hóa hơn. Các nghiên cứu tương lai nên tập trung vào giải quyết những thách thức hiện tại, tích hợp thông tin bổ sung về ngữ cảnh, tăng cường tính minh bạch và đảm bảo khả năng mở rộng để làm cho những hệ thống này trở nên hiệu quả và áp dụng trong các tình huống thực tế.*

***Cảm ơn các thầy và tất cả các bạn đã lắng nghe!!!!***