

TỔNG CÔNG TY VIỄN THÔNG VIETTEL

KHỐI CÔNG NGHỆ THÔNG TIN

**PYC TRIỂN KHAI CHO XÂY DỰNG GIẢI PHÁP CHO BÀI TOÁN RECOMMENDATION ITEMS**

**TÀI LIỆU MÔ TẢ YÊU CẦU**

Mã hiệu dự án: MyViettel\_0009

Mã hiệu tài liệu: MyViettel\_0009

<Địa điểm, Thời gian>

BẢNG GHI NHẬN TIẾN ĐỘ

\*A - Tạo mới, M - Sửa đổi, D - Xóa bỏ

| Ngày  bàn giao | Yêu cầu | A\*  M, D | Trạng thái | Đầu mối KH | Mô tả | Ghi chú |
| --- | --- | --- | --- | --- | --- | --- |
| 19/5/2023 | Xây dựng giải pháp cho bài toán recommend items | A | Hoàn thành | PTC, PKH | Xây dựng giải pháp mô hình, sắp xếp items trong từng nhóm tiện ích của app myvt |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

MỤC LỤC

[I. Mở đầu 6](#_heading=h.2et92p0)

[1.1 Khảo sát bài toán recommendation ranking 6](#_heading=h.cylagozjj9j)

[1.1.1 Mô-đun xếp hạng trong hệ thống gợi ý 6](#_heading=h.858x4lfdfmv8)

[1.1.2 Giới thiệu về thuật toán xếp hạng 7](#_heading=h.gt3beii0ahox)

[**II. Đề xuất xây dựng mô hình 8**](#_heading=h.43s3suuam4hp)

[2.1 Đề xuất các mô hình 8](#_heading=h.xvlpxnpsqtq1)

[2.1.1 Xếp hạng lại được cá nhân hóa cho gợi ý 9](#_heading=h.5vu56446qpy8)

[2.1.2 DCN V2: Cải thiện mạng sâu & chéo và các bài học thực tế cho hệ thống học tập trên quy mô web để xếp hạng 10](#_heading=h.6ovta81ho2px)

[2.2 Thống nhất lựa chọn mô hình 11](#_heading=h.9295fwxt87kv)

[2.3 Personalized Re-ranking for Recommendation 11](#_heading=h.2iqrbmhh9xzs)

[2.3.1 Công thức mô hình re-ranking: 11](#_heading=h.yt55niw0rrlg)

[2.3.2 PERSONALIZED RE-RANKING MODEL 12](#_heading=h.te5sk4iiwvg7)

[2.3.2.1 Tổng quan kiến trúc mô hình 12](#_heading=h.e0ng2qgmwpfs)

[2.3.2.2 Input Layer 13](#_heading=h.t276t8wfw1u)

[2.3.2.3 Encoding Layer 14](#_heading=h.pi4i19oy0e2p)

[2.3.2.4 Output Layer 15](#_heading=h.dvi06wrf4o2v)

[2.3.2.5 Personalized Module 16](#_heading=h.5e9pxd902nj0)

[2.3.3 Baselines 17](#_heading=h.oa4v11l5hj58)

[2.4 DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems 18](#_heading=h.a2yxergi57kw)

[2.4.1 Tổng quan 18](#_heading=h.qa3v17trxrc5)

[2.4.1.1 Mô tả chung 18](#_heading=h.55ooopn5ud2g)

[2.4.1.2 Learn to rank(LTR) 19](#_heading=h.f13603fwxmdw)

[2.4.2 Khảo sát các mô hình khác 20](#_heading=h.5fxpandce07b)

[2.4.2.1 Wide & Deep Learning for Recommender Systems 20](#_heading=h.frb4c6b575a1)

[2.4.2.1.1 Thành phần rộng 21](#_heading=h.ov9jwii04p6c)

[2.4.2.1.2 Thành phần sâu 21](#_heading=h.dfcjey3uwvxi)

[2.4.2.2 Interpretable Click-Through Rate Prediction through Hierarchical Attention 22](#_heading=h.o1uj5yfakjgc)

[2.4.2.2.1 Multi-head Transformer 23](#_heading=h.qz4zfu6t5x3u)

[2.4.2.2 Hierarchical Attention 24](#_heading=h.f4622mvmisug)

[2.4.2.3 Evaluating collaborative filtering recommender systems 25](#_heading=h.qusqfgyuni2y)

[2.4.2.3.1 MEMORY-BASED COLLABORATIVE FILTERING 26](#_heading=h.exurg98738wk)

[2.4.2.3.2 MODEL-BASED COLLABORATIVE FILTERING 27](#_heading=h.1ya5l1k1xx73)

[2.4.3 Cấu trúc song song và cấu trúc xếp chồng 28](#_heading=h.ljjjnr56suy5)

[2.4.3.1 Cấu trúc song song 28](#_heading=h.26epd7fhip8)

[2.4.3.2 Cấu trúc xếp chồng 29](#_heading=h.508fvvhm3wc4)

[2.4.4 Kiến trúc mô hình 31](#_heading=h.k71mtsk1xhvl)

[2.4.4.1 Embedding Layer 31](#_heading=h.97vc2me5bb2h)

[2.4.4.2 Cross Network 32](#_heading=h.msf4l5q152kq)

[2.4.4.3 Deep Network 33](#_heading=h.2tnottw8zbsm)

[2.4.4.4 Deep and Cross Combination 34](#_heading=h.621oz131erj8)

[2.4.4.5 Cost-Effective Mixture of Low-Rank DCN 35](#_heading=h.ip4ziar9wjfo)

[2.4.4.6 Complexity Analysis 37](#_heading=h.2r9usr5n45aj)

[2.4.5 Kỹ thuật chéo đặc trưng 38](#_heading=h.waf8vu8wvdo8)

[2.4.5.1 Hiệu suất với độ khó tăng dần 38](#_heading=h.iqra1jcjca3x)

[2.4.5.2 Vai trò của từng thành phần 39](#_heading=h.uzehswjxo3t6)

[2.4.5.3 Hiệu suất với độ sâu lớp ngày càng tăng 39](#_heading=h.sy1bn0axxth2)

[2.4.6 Phân tích mô hình 40](#_heading=h.1x12sm8i9niz)

[2.4.6.1 Xấp xỉ đa thức 40](#_heading=h.b34nfmldk7n7)

[2.4.6.2 Kết nối với công việc liên quan 41](#_heading=h.48w2rn2hjnpg)

TRANG KÝ

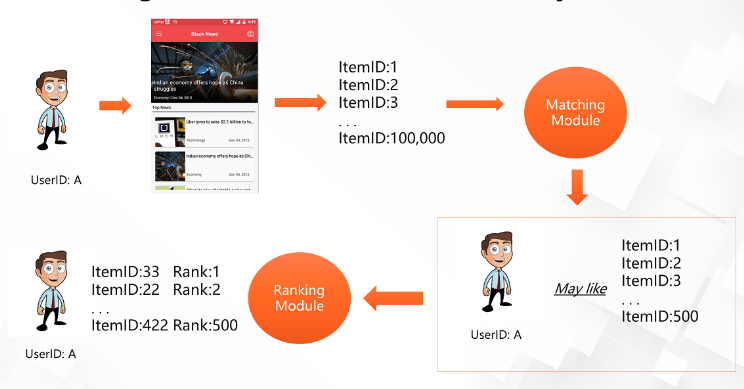
# **Mở đầu**

## 1.1 Khảo sát bài toán recommendation ranking

### 1.1.1 Mô-đun xếp hạng trong hệ thống gợi ý

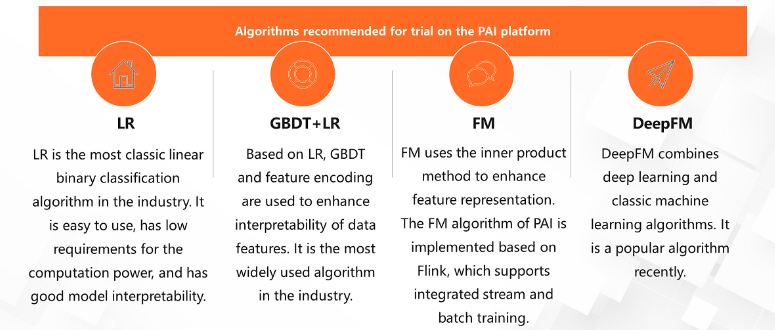
Các thuật toán xếp hạng của một hệ thống tư vấn và các tính năng của kiến ​​trúc đào tạo trực tuyến và ngoại tuyến cho một mô hình xếp hạng. Trước tiên, hãy xem lại hình sau để tìm hiểu về mô-đun xếp hạng trong hệ thống tư vấn. Khi người dùng truy cập một nền tảng, người dùng sẽ tìm thấy một số lượng lớn các mặt hàng. Do đó, chúng ta cần lọc ra những mục mà người dùng có thể thích. Mô-đun so khớp cung cấp bộ lọc sơ bộ để thu hẹp danh sách mặt hàng được gửi đến mô-đun xếp hạng.

Cụ thể, mô-đun so khớp trước tiên chọn một tỷ lệ nhỏ các mục mà người dùng có thể thích từ một số lượng lớn các mục. Ví dụ: nền tảng có 100.000 mặt hàng và mô-đun phù hợp sẽ lọc ra 500 mặt hàng mà người dùng có thể thích. Sau đó, mô-đun xếp hạng xếp hạng các mục này dựa trên sở thích của người dùng. Để xếp hạng 500 mục từ yêu thích của người dùng đến ít yêu thích nhất, chúng tôi cần một thuật toán xếp hạng, có thể cung cấp thứ hạng của các mục dựa trên thuộc tính của người dùng và mục.



### 1.1.2 Giới thiệu về thuật toán xếp hạng

Phần này mô tả các loại thuật toán xếp hạng, quy trình đào tạo mô hình xếp hạng và kiến ​​trúc đào tạo. Với sự phát triển của học sâu, các thuật toán xếp hạng đã dần được tích hợp với học sâu. Hình dưới đây cho thấy các thuật toán xếp hạng điển hình. Đầu tiên là hồi quy logistic (LR), đây là một thuật toán được sử dụng rộng rãi. Đây là thuật toán nhị phân tuyến tính cổ điển nhất trong ngành. Nó dễ sử dụng, có yêu cầu thấp về sức mạnh tính toán và có khả năng diễn giải mô hình tốt. Thứ hai là Máy nhân tố hóa (FM), đã được áp dụng cho một số lượng lớn các tình huống của khách hàng trong một hoặc hai năm qua và đạt được hiệu suất tốt. Nó sử dụng phương pháp sản phẩm bên trong để tăng cường biểu diễn tính năng. Thứ ba là LR sử dụng cây quyết định tăng cường độ dốc (GBDT) và mã hóa tính năng để nâng cao khả năng diễn giải của các tính năng dữ liệu. Thứ tư là thuật toán DeepFM, đây cũng là một thuật toán học sâu được sử dụng rộng rãi. Nó kết hợp các thuật toán học sâu và học máy cổ điển. Nếu bạn cố gắng xây dựng một hệ thống gợi ý lần đầu tiên, chúng tôi khuyên bạn trước tiên nên thử các thuật toán đơn giản và sau đó sử dụng các thuật toán phức tạp.



1.1.3 **Kiến trúc đào tạo của mô hình xếp hạng**

Phần này mô tả cách bạn có thể đào tạo một mô hình xếp hạng. Một mô hình xếp hạng có thể được đào tạo trực tuyến hoặc ngoại tuyến và trước tiên chúng tôi sẽ giới thiệu đào tạo ngoại tuyến.

Đào tạo ngoại tuyến sử dụng dữ liệu T – 1. Cụ thể, các mô hình được sử dụng cho các doanh nghiệp ngày nay được đào tạo dựa trên dữ liệu từ trước ngày hôm nay. Đào tạo ngoại tuyến cho phép bạn tích hợp một lượng lớn dữ liệu lịch sử vào một kho dữ liệu, đào tạo mô hình dựa trên các tính năng và dữ liệu của tất cả các mẫu, đồng thời xác minh mô hình được đào tạo ngoại tuyến. Sau đó, bạn có thể sử dụng mô hình đã được xác minh trực tuyến vào ngày hôm sau, điều này đảm bảo tính bảo mật và hiệu suất tốt. Hình dưới đây cung cấp kiến ​​trúc đào tạo mô hình ngoại tuyến. Hiện nay, hầu hết khách hàng đào tạo xếp hạng mô hình nhé. Nếu bạn muốn thực hiện đào tạo thời gian thực, bạn sẽ phải đối mặt với những thách thức lớn trong kiến ​​trúc.

# **Đề xuất xây dựng mô hình**

## 2.1 Đề xuất các mô hình

Mô hình recommend ranking và learn to rank là những mô hình học máy được sử dụng để xếp hạng các mục theo mức độ liên quan hoặc thích hợp với một truy vấn hoặc một người dùng. Một số ứng dụng của các mô hình này là tìm kiếm web, gợi ý sản phẩm, gợi ý tin tức, gợi ý bài viết, v.v.

Các mô hình recommend ranking thường sử dụng các thuật toán lọc cộng tác (collaborative filtering) hoặc lọc nội dung (content-based filtering) để tạo ra các danh sách gợi ý cho người dùng dựa trên sở thích, hành vi hoặc đặc điểm của họ. Các thuật toán lọc cộng tác sử dụng các thông tin về sự tương tác giữa người dùng và các mục để tìm ra những mục tương tự hoặc những người dùng có sở thích giống nhau. Các thuật toán lọc nội dung sử dụng các thông tin về nội dung của các mục để tìm ra những mục phù hợp với sở thích hoặc yêu cầu của người dùng.

Các mô hình learn to rank thường sử dụng các thuật toán học có giám sát (supervised learning) để tối ưu hóa một hàm xếp hạng dựa trên các đặc trưng của cặp truy vấn-mục và các nhãn xếp hạng được cung cấp bởi người đánh giá. Các thuật toán learn to rank có thể được phân loại thành ba loại chính: point-wise models, pairwise models và listwise models. Các point-wise models dự đoán một điểm số đối với từng cặp truy vấn-mục trong tập dữ liệu, và sử dụng nó để xếp hạng các mục. Các pairwise models so sánh hai cặp truy vấn-mục và dự đoán cái nào có thứ hạng cao hơn. Các listwise models xem xét toàn bộ danh sách các cặp truy vấn-mục và dự đoán thứ tự tối ưu của chúng.

### 2.1.1 Xếp hạng lại được cá nhân hóa cho gợi ý

Xếp hạng là một nhiệm vụ cốt lõi trong các hệ thống giới thiệu, nhằm mục đích cung cấp một danh sách các mặt hàng được sắp xếp theo thứ tự cho người dùng. Thông thường, một chức năng xếp hạng được học từ tập dữ liệu được gắn nhãn để tối ưu hóa hiệu suất toàn cầu, tạo ra điểm xếp hạng cho từng mục riêng lẻ. Tuy nhiên, nó có thể không tối ưu vì chức năng tính điểm áp dụng cho từng mục riêng lẻ và không xem xét rõ ràng ảnh hưởng lẫn nhau giữa các mục, cũng như sự khác biệt về sở thích hoặc ý định của người dùng. do đó, chúng tôi đề xuất một mô hình xếp hạng lại được cá nhân hóa cho các hệ thống đề xuất. Mô hình xếp hạng lại được đề xuất có thể dễ dàng triển khai dưới dạng mô-đun tiếp theo sau bất kỳ thuật toán xếp hạng nào, bằng cách sử dụng trực tiếp các vectơ đặc trưng xếp hạng hiện có. Nó trực tiếp tối ưu hóa toàn bộ danh sách đề xuất bằng cách sử dụng cấu trúc biến áp để mã hóa thông tin của tất cả các mục trong danh sách một cách hiệu quả. Cụ thể, Transformer áp dụng cơ chế tự định hướng trực tiếp mô hình hóa các mối quan hệ toàn cầu giữa bất kỳ cặp mục nào trong toàn bộ danh sách. Chúng tôi xác nhận rằng hiệu suất có thể được cải thiện hơn nữa bằng cách giới thiệu tính năng nhúng được đào tạo trước để tìm hiểu các chức năng mã hóa được cá nhân hóa cho những người dùng khác nhau. Kết quả thử nghiệm trên cả điểm chuẩn oine và hệ thống thương mại điện tử trực tuyến trong thế giới thực cho thấy những cải tiến đáng kể của mô hình xếp hạng lại được đề xuất.

### 2.1.2 DCN V2: Cải thiện mạng sâu & chéo và các bài học thực tế cho hệ thống học tập trên quy mô web để xếp hạng

Tìm hiểu các tính năng chéo hiệu quả là chìa khóa đằng sau việc xây dựng hệ thống đề xuất. Tuy nhiên, không gian tính năng thưa thớt và rộng lớn đòi hỏi phải tìm kiếm toàn diện để xác định các phép lai hiệu quả. Mạng Deep & Cross (DCN) đã được đề xuất để học tự động và hiệu quả các tương tác tính năng dự đoán mức độ giới hạn. Thật không may, trong các mô hình phục vụ lưu lượng truy cập quy mô web với hàng tỷ ví dụ đào tạo, DCN đã cho thấy khả năng diễn đạt hạn chế trong mạng chéo của mình khi tìm hiểu thêm các tương tác tính năng dự đoán. Mặc dù đã đạt được những tiến bộ đáng kể trong nghiên cứu, nhưng nhiều mô hình học sâu trong sản xuất vẫn dựa vào các mạng thần kinh chuyển tiếp nguồn cấp dữ liệu truyền thống để tìm hiểu các đặc điểm chéo một cách không hiệu quả. Dựa trên những ưu/nhược điểm của DCN và các phương pháp học tập tương tác tính năng hiện có, chúng tôi đề xuất một khung cải tiến DCN-V2 để làm cho DCN trở nên thiết thực hơn trong môi trường công nghiệp quy mô lớn. Trong một nghiên cứu thử nghiệm toàn diện với việc điều chỉnh mô hình và tìm kiếm siêu tham số mở rộng, chúng tôi đã quan sát thấy rằng các phương pháp tiếp cận DCN-V2 vượt trội hơn tất cả các thuật toán tiên tiến nhất trên các bộ dữ liệu điểm chuẩn phổ biến. DCN-V2 cải tiến mang tính biểu cảm cao hơn nhưng vẫn tiết kiệm chi phí khi học tương tác tính năng, đặc biệt khi được kết hợp với sự kết hợp của kiến ​​trúc cấp thấp. DCN-V2 rất đơn giản, có thể dễ dàng sử dụng làm các khối xây dựng và đã mang lại độ chính xác ngoại tuyến đáng kể và các chỉ số kinh doanh trực tuyến đạt được trên nhiều hệ thống học tập quy mô web để xếp hạng tại Google.

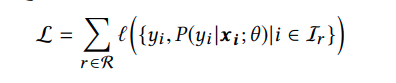
## 2.2 Thống nhất lựa chọn mô hình

Dựa trên sự phân tích bài toán và các mô hình hiện có họ thống nhất các giải pháp xây dựng các mô hình sau: Personalized Re-ranking for Recommendation, DCNv2

## 2.3 Personalized Re-ranking for Recommendation

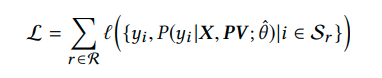
### 2.3.1 Công thức mô hình re-ranking:

Phương pháp học cách xếp hạng (được gắn nhãn là LTR) được sử dụng rộng rãi để xếp hạng trong các hệ thống làm việc thực nhằm tạo ra một danh sách có thứ tự để truy xuất thông tin và đề xuất. Phương pháp LTR học một chức năng chấm điểm toàn cầu dựa trên vectơ đặc trưng của các mục. Có chức năng toàn cục này, phương thức LTR xuất ra một danh sách có thứ tự bằng cách cho điểm từng mục trong tập ứng viên. là hàm tính điểm toàn cục thường được học bằng cách giảm thiểu hàm mất mát L sau:



trong đó R là tập hợp tất cả các yêu cầu đề xuất của người dùng. Ir là ​​tập ứng viên của các mục cho yêu cầu r ∈ R. xi đại diện cho không gian đặc trưng của mục i. yi là nhãn trên mục i, tức là có nhấp chuột hay không. P(yi |xi ; θ) là xác suất nhấp chuột dự đoán của mục i được đưa ra bởi mô hình xếp hạng với tham số θ. ` là tổn thất được tính với yi và P(yi |xi ; θ).

Tuy nhiên, xi không đủ để học một chức năng tính điểm tốt. Chúng tôi thấy rằng xếp hạng cho hệ thống tư vấn nên xem xét các thông tin bổ sung sau: (a) ảnh hưởng lẫn nhau giữa các cặp vật phẩm; (b) tương tác giữa người dùng và các mặt hàng. Có thể học trực tiếp ảnh hưởng lẫn nhau giữa các cặp vật phẩm từ danh sách ban đầu Sr = [i1,i2, ...,in] do mô hình LTR hiện có cho yêu cầu r đưa ra. Works đề xuất các cách tiếp cận để tận dụng tốt hơn thông tin chung của các cặp vật phẩm. Tuy nhiên, rất ít công trình xem xét các tương tác giữa người dùng và vật phẩm. Mức độ ảnh hưởng lẫn nhau của các cặp vật phẩm thay đổi tùy theo người dùng. Trong bài báo này, chúng tôi giới thiệu một ma trận PV được cá nhân hóa để tìm hiểu chức năng mã hóa dành riêng cho người dùng, chức năng này có thể mô hình hóa các ảnh hưởng lẫn nhau được cá nhân hóa giữa các cặp vật phẩm. Hàm mất mát của mô hình có thể được xây dựng như:

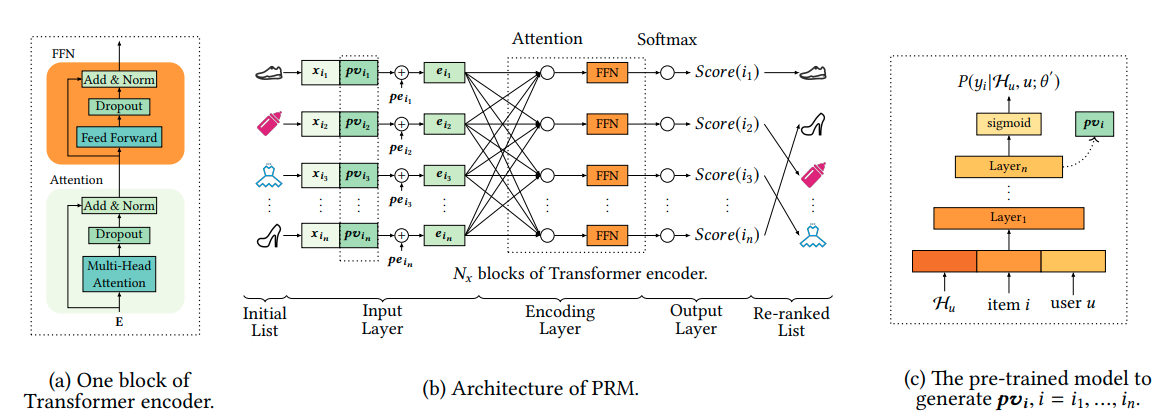


trong đó Sr là danh sách ban đầu được đưa ra bởi mô hình xếp hạng trước đó. ˆθ là tham số của mô hình xếp hạng lại của chúng tôi. X là ma trận đặc trưng của tất cả các mục trong danh sách.

### 2.3.2 PERSONALIZED RE-RANKING MODEL

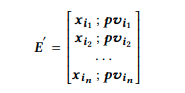
#### 2.3.2.1 Tổng quan kiến trúc mô hình

Kiến trúc của mô hình PRM được thể hiện trong hình. mô hình bao gồm ba phần: lớp đầu vào, lớp mã hóa và lớp đầu ra. Nó lấy danh sách ban đầu của các mục được tạo bởi phương pháp xếp hạng trước đó làm đầu vào và xuất ra một danh sách được xếp hạng lại. cấu trúc chi tiết sẽ được giới thiệu riêng trong các phần sau.

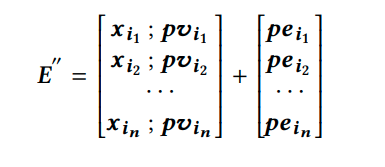


#### 2.3.2.2 Input Layer

Mục tiêu của lớp đầu vào là chuẩn bị các biểu diễn toàn diện của tất cả các mục trong danh sách ban đầu và cung cấp nó cho lớp mã hóa. Đầu tiên, chúng ta có độ dài xed của danh sách tuần tự ban đầu S = [i1,i2, ...,in] được đưa ra bởi phương pháp xếp hạng trước đó. Tương tự như phương pháp xếp hạng trước, chúng ta có ma trận đặc trưng thô X ∈ R n×d  
(feature) . Mỗi hàng trong X đại diện cho véc tơ đặc trưng thô xi cho mỗi mục i ∈ S. Vectơ cá nhân hóa (PV ). Việc mã hóa các vectơ đặc trưng của hai mục có thể mô hình hóa các ảnh hưởng lẫn nhau giữa chúng, nhưng những ảnh hưởng này có thể ảnh hưởng đến người dùng ở mức độ nào thì không rõ. Cần phải học chức năng mã hóa dành riêng cho người dùng. Mặc dù việc thể hiện toàn bộ danh sách ban đầu có thể phản ánh một phần tùy chọn của người dùng, nhưng điều đó là không đủ đối với chức năng mã hóa được cá nhân hóa mạnh mẽ. Như được hiển thị trong Hình 1 (b), chúng tôi nối ma trận tính năng thô X ∈ R n×d(feature) với ma trận được cá nhân hóa PV ∈ R n×dpv để có được ma trận nhúng trung gian E 0 ∈ R n×(d(feature)+dpv ) , trong đó được thể hiện trong Phương trình 3. PV được tạo ra bởi một mô hình được đào tạo trước sẽ được giới thiệu trong phần sau. Hiệu suất đạt được của PV sẽ được giới thiệu trong phần đánh giá.



Nhúng vị trí (PE). Để sử dụng thông tin tuần tự trong danh sách ban đầu, chúng tôi đưa một vị trí nhúng PE ∈ R n×(feature+dpv ) vào phần nhúng đầu vào. en ma trận nhúng cho lớp mã hóa có thể được tính bằng Công thức 4. Trong bài báo này, một PE có thể học được được sử dụng mà chúng tôi nhận thấy rằng nó vượt trội hơn một chút so với nhúng vị trí được sử dụng trong đó.

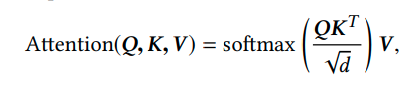


Cuối cùng, chúng tôi sử dụng một mạng chuyển tiếp nguồn cấp dữ liệu đơn giản để chuyển đổi ma trận đặc trưng E 00 ∈ R n×(dfeature+dpv ) thành E ∈ R n×d , trong đó d là chiều tiềm ẩn của mỗi vectơ đầu vào của lớp mã hóa. E có thể được xây dựng như:

trong đó W E ∈ R (dfeature+dpv )×d là ma trận chiếu và b E là vectơ d chiều.

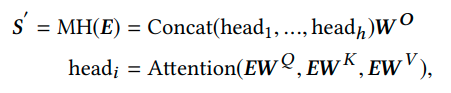
#### 2.3.2.3 Encoding Layer

Mục tiêu của lớp mã hóa trong Hình 1(a) là tích hợp ảnh hưởng lẫn nhau của các cặp vật phẩm và thông tin bổ sung khác, bao gồm sở thích của người dùng và thứ tự xếp hạng của danh sách ban đầu S. Để đạt được mục tiêu này, chúng tôi sử dụng bộ mã hóa giống Transformer vì Transformer đã được chứng minh là hiệu quả trong nhiều tác vụ NLP, đặc biệt trong dịch máy nhờ khả năng mã hóa và giải mã mạnh mẽ so với các phương pháp dựa trên RNN. Cơ chế tự xác định trong Transformer đặc biệt phù hợp trong nhiệm vụ xếp hạng lại của chúng tôi vì nó trực tiếp mô hình hóa các ảnh hưởng lẫn nhau đối với hai mục bất kỳ bất kể khoảng cách giữa chúng. Không phân rã khoảng cách, Transformer có thể ghi lại nhiều tương tác hơn giữa các mục cách xa nhau trong danh sách ban đầu. Như được hiển thị, mô-đun mã hóa của chúng tôi bao gồm Nx khối của bộ mã hóa Biến áp. Mỗi khối chứa một lớp attention và lớp Mạng chuyển tiếp nguồn cấp dữ liệu (FFN).



trong đó các ma trận Q, K, V tương ứng đại diện cho các truy vấn, khóa và giá trị. d là số chiều của ma trận K để tránh giá trị lớn của tích bên trong. somax được sử dụng để chuyển đổi giá trị của tích bên trong thành trọng số cộng của vectơ giá trị V . Trong bài báo của chúng tôi, chúng tôi sử dụng phép tự định hình trong đó Q, K và V được chiếu từ cùng một ma trận.

Để mô hình hóa các ảnh hưởng lẫn nhau phức tạp hơn, chúng ta sử dụng quan điểm nhiều đầu như hình minh họa:



trong đó W Q ,W K ,WV ∈ R d×d . W O ∈ R hd×d model là ma trận chiếu. h là số lượng tiêu đề. e ảnh hưởng của giá trị khác nhau của h sẽ được nghiên cứu trong nghiên cứu cắt bỏ trong phần tiếp theo.

Mạng chuyển tiếp nguồn cấp dữ liệu. Chức năng của Mạng chuyển tiếp nguồn cấp dữ liệu (FFN) khôn ngoan theo vị trí này chủ yếu là để cải thiện mô hình với tính phi tuyến tính và tương tác giữa các kích thước khác nhau của vectơ đầu vào.

Xếp chồng lớp mã hóa. Ở đây, chúng tôi sử dụng mô-đun attention theo sau là FFN thông minh về vị trí như một khối của bộ mã hóa Transformer. Bằng cách xếp chồng nhiều khối, chúng ta có thể nhận được thông tin lẫn nhau phức tạp hơn và có thứ tự cao hơn.

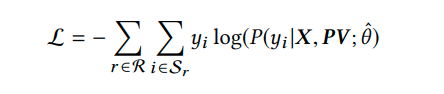
#### 2.3.2.4 Output Layer

Chức năng của lớp đầu ra chủ yếu là tạo điểm cho từng mục i = i1, . . . ,in (được gắn nhãn là Điểm(i) trong Hình 1(b)). Chúng tôi sử dụng một lớp tuyến tính theo sau là lớp softmax. đầu ra của lớp softmax là xác suất nhấp chuột cho từng mục, được gắn nhãn là P(yi |X, PV ; ˆθ). Chúng tôi sử dụng P(yi |X, PV ; ˆθ) làm Điểm(i) để xếp hạng lại các mục trong một bước. công thức của Điểm (i) là:



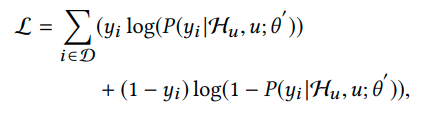
trong đó F (Nx ) là đầu ra của khối Nx của bộ mã hóa Biến áp. WF là ma trận chiếu có thể học được và b F là thuật ngữ sai lệch. n là số phần tử trong danh sách ban đầu.

Trong quá trình đào tạo, chúng tôi sử dụng dữ liệu nhấp qua làm nhãn và giảm thiểu hàm mất mát được hiển thị trong Công thức.



#### 2.3.2.5 Personalized Module

Trong phần này, chúng tôi giới thiệu phương pháp tính toán ma trận PV được cá nhân hóa, đại diện cho các tương tác giữa người dùng và vật phẩm. Cách tiếp cận đơn giản là tìm hiểu PV với mô hình PRM theo cách từ đầu đến cuối thông qua tổn thất xếp hạng lại. Tuy nhiên, như đã giải thích trong Phần 3, nhiệm vụ xếp hạng lại là làm mới kết quả của các phương pháp xếp hạng trước đó. Mô tả nhiệm vụ cụ thể được học khi xếp hạng lại nhiệm vụ thiếu tùy chọn chung của người dùng. do đó, chúng tôi sử dụng mạng thần kinh được đào tạo trước để tạo ra PV nhúng được cá nhân hóa của người dùng, sau đó được sử dụng làm tính năng bổ sung cho mô hình PRM. Mạng thần kinh được đào tạo trước được học từ toàn bộ nhật ký nhấp quả của nền tảng. Hình 1 (c) cho thấy cấu trúc của mô hình được đào tạo trước được sử dụng trong bài báo của chúng tôi. lớp sigmoid đưa ra xác suất nhấp chuột (P(yi |Hu,u; θ 0 )) trên mục i cho người dùng u với lịch sử tất cả hành vi của người dùng (Hu ) và thông tin phụ của người dùng. Thông tin bên điện tử của người dùng bao gồm giới tính, độ tuổi và mức độ mua hàng, et.al. e mất mát của mô hình được tính bằng hàm entropy chéo theo điểm được hiển thị



trong đó D là tập hợp các mục được hiển thị cho người dùng u trên nền tảng. θ 0 là ma trận tham số của mô hình được đào tạo trước. yi là nhãn (nhấp chuột hoặc không) trên mục i. Lấy cảm hứng từ công việc [13], chúng tôi sử dụng vectơ ẩn trước lớp sigmoid làm pvi vectơ được cá nhân hóa (trong Hình 1(c)) cung cấp cho mô hình PRM của chúng tôi. Hình 1(c) cho thấy một kiến ​​trúc có thể có của mô hình được đào tạo trước, các mô hình chung khác như FM, FFM, DeepFM, DCN, FNN và PNN cũng có thể được sử dụng làm phương án thay thế để tạo ra PV.

### 2.3.3 Baselines

Cả phương pháp học cách xếp hạng (LTR) và xếp hạng lại đều có thể đóng vai trò là cơ sở của chúng tôi.

**LTR**. Các phương pháp LTR được sử dụng trong hai nhiệm vụ. Thứ nhất, các phương pháp LTR có thể tạo ra một danh sách ban đầu Sr cho mô hình xếp hạng lại từ một tập hợp ứng cử viên Ir cho mỗi yêu cầu r của người dùng. Thứ hai, các phương pháp LTR sử dụng hàm mất mát theo cặp hoặc theo danh sách có thể hoạt động như các phương pháp xếp hạng lại bằng cách lấy danh sách ban đầu Sr làm đầu vào và tiến hành thuật toán xếp hạng trong thời gian khác. Các phương pháp LTR đại diện được sử dụng trong bài viết này bao gồm:

• SVMRank]: Đây là một phương pháp học xếp hạng tiêu biểu sử dụng tổn thất theo cặp để mô hình hóa chức năng tính điểm.

• LambdaMart: Đây là phương pháp học xếp hạng đại diện sử dụng tổn thất theo danh sách để lập mô hình hàm tính điểm. LambdaMart là LTR tiên tiến nhất trong số các phương pháp LTR đó được trang bị hàm mất mát theo danh sách theo đánh giá của.

• LTR dựa trên DNN: Đây là phương pháp học để xếp hạng được triển khai trong hệ thống đề xuất trực tuyến của chúng tôi. Nó sử dụng cấu trúc mạng Wide&Deep tiêu chuẩn để lập mô hình chức năng tính điểm thông qua chức năng mất mát theo từng điểm.

**Re-ranking**. Như đã đề cập trong phần công việc liên quan, các phương pháp xếp hạng lại hiện có bao gồm DLCM, Seq2Slate và GlobalRerank. DLCM[1] và GlobalRerank tập trung vào xếp hạng lại trong truy xuất thông tin. Seq2Slate tập trung vào xếp hạng lại trong cả đề xuất và truy xuất thông tin. Seq2Slate và GlobalRerank không được chọn làm đường cơ sở vì tất cả chúng đều sử dụng cấu trúc bộ giải mã để tạo danh sách được xếp hạng lại. Seq2Slate sử dụng mạng con trỏ để tạo danh sách được xếp hạng lại theo tuần tự. GlobalRerank sử dụng RNN được trang bị cơ chế ention làm bộ giải mã. Cấu trúc bộ giải mã xuất từng mục một. Việc một mục có được chọn hay không phụ thuộc vào các mục được chọn trước nó. Do đó, cả Seq2Slate và GlobalRerank đều không thể song song trong suy luận trực tuyến. độ phức tạp về thời gian cho Seq2Slate và GlobalRerank ở giai đoạn giao thoa là O(n) × RT , trong đó n là độ dài của danh sách ban đầu và RT là thời gian cho một yêu cầu xếp hạng hoặc xếp hạng lại. Độ trễ để xếp hạng lại bởi Seq2Slate và GlobalRerank là không thể chấp nhận được do tiêu chí độ trễ nghiêm ngặt đối với dịch vụ đề xuất trực tuyến.

• DLCM: Đây là mô hình xếp hạng lại được sử dụng trong truy xuất thông tin dựa trên danh sách ban đầu do phương pháp LTR tạo ra. GRU được sử dụng để mã hóa thông tin ngữ cảnh cục bộ thành một vectơ toàn cục. Kết hợp vectơ tổng thể và từng vectơ đặc trưng, ​​nó học được một chức năng tính điểm mạnh mẽ hơn so với chức năng xếp hạng toàn cầu của LTR.

## 2.4 DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems

### 2.4.1 Tổng quan

#### 2.4.1.1 Mô tả chung

Học tập các phép lai tính năng hiệu quả là chìa khóa đằng sau việc xây dựng hệ thống tư vấn. Tuy nhiên, không gian đối tượng địa lý rộng và thưa thớt đòi hỏi phải tìm kiếm đầy đủ để xác định các dấu thập hiệu quả. Deep & Cross Network (DCN) đã được đề xuất để tìm hiểu tự động và hiệu quả các tương tác của tính năng dự đoán ở mức độ giới hạn. Thật không may, trong các mô hình phục vụ lưu lượng truy cập quy mô web với hàng tỷ ví dụ đào tạo, DCN cho thấy khả năng biểu đạt hạn chế trong mạng chéo của nó khi tìm hiểu thêm các tương tác tính năng dự đoán. Bất chấp những tiến bộ nghiên cứu đáng kể đã đạt được, nhiều mô hình học sâu trong sản xuất vẫn dựa vào mạng nơ-ron truyền về phía trước truyền thống để tìm hiểu các tính năng không hiệu quả.

Dựa trên những ưu / nhược điểm của DCN và các phương pháp tiếp cận học tập tương tác tính năng hiện có, chúng tôi đề xuất một khuôn khổ cải tiến DCN-V2 để làm cho DCN thiết thực hơn trong các cơ sở công nghiệp quy mô lớn. Trong một nghiên cứu thử nghiệm toàn diện với tìm kiếm siêu tham số và điều chỉnh mô hình, chúng tôi nhận thấy rằng phương pháp tiếp cận DCN-V2 hoạt động tốt hơn tất cả các thuật toán hiện đại trên các bộ dữ liệu điểm chuẩn phổ biến. DCN-V2 được cải tiến mang tính biểu cảm hơn nhưng vẫn tiết kiệm chi phí trong quá trình học tương tác tính năng, đặc biệt là khi kết hợp với sự kết hợp của kiến ​​trúc cấp thấp. DCN-V2 rất đơn giản, có thể dễ dàng được sử dụng như các khối xây dựng và đã mang lại độ chính xác ngoại tuyến đáng kể và các chỉ số kinh doanh trực tuyến đạt được qua nhiều quy mô web để xếp hạng các hệ thống tại Google.

#### 2.4.1.2 Learn to rank(LTR)

Học cách xếp hạng (LTR)vẫn là một trong những vấn đề quan trọng nhất trong học máy hiện đại và học sâu. Nó có một loạt các ứng dụng trong hệ thống tìm kiếm, đề xuất và quảng cáo tính toán. Trong số các thành phần quan trọng của mô hình LTR, việc học các phép lai tính năng hiệu quả tiếp tục thu hút rất nhiều sự chú ý từ cả giới học thuật và ngành công nghiệp.

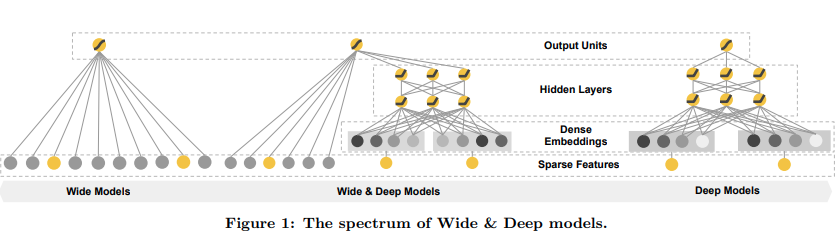
Học cách xếp hạng rất hữu ích cho việc truy xuất tài liệu, lọc cộng tác và nhiều ứng dụng khác. Một số phương pháp để học xếp hạng đã được đề xuất, trong đó sử dụng các cặp đối tượng làm 'ví dụ' trong học tập. Chúng tôi gọi chúng là cách tiếp cận theo cặp trong bài báo này. Mặc dù cách tiếp cận theo cặp mang lại lợi ích, nhưng nó bỏ qua thực tế rằng xếp hạng là một nhiệm vụ dự đoán trên danh sách các đối tượng. Bài báo giả định rằng việc học xếp hạng nên áp dụng phương pháp tiếp cận theo danh sách, trong đó danh sách các đối tượng được sử dụng làm 'cá thể' trong học tập. Bài báo đề xuất một phương pháp xác suất mới cho cách tiếp cận. Cụ thể, nó giới thiệu hai mô hình xác suất, tương ứng được gọi là xác suất hoán vị và xác suất cao nhất, để xác định một hàm mất mát theo danh sách cho việc học. Neural Network và Gradient Descent sau đó được sử dụng làm mô hình và thuật toán trong phương pháp học tập. Kết quả thử nghiệm về truy xuất thông tin cho thấy rằng cách tiếp cận theo danh sách được đề xuất hoạt động tốt hơn so với cách tiếp cận theo cặp.

Học cách xếp hạng, khi được áp dụng cho việc truy xuất tài liệu, là một nhiệm vụ như sau. Giả sử rằng có một bộ sưu tập các tài liệu. Trong truy xuất (tức là xếp hạng), khi đưa ra một truy vấn, chức năng xếp hạng sẽ ấn định điểm cho mỗi tài liệu và xếp hạng các tài liệu theo thứ tự giảm dần của điểm. Thứ tự xếp hạng thể hiện mức độ liên quan tương đối của các tài liệu đối với truy vấn. Trong học tập, một số truy vấn được cung cấp; mỗi truy vấn được liên kết với một danh sách xếp hạng hoàn hảo của các tài liệu; một chức năng xếp hạng sau đó được tạo bằng cách sử dụng dữ liệu đào tạo, để mô hình có thể dự đoán chính xác các danh sách xếp hạng trong dữ liệu đào tạo.

### 2.4.2 Khảo sát các mô hình khác

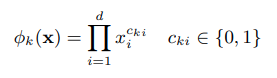
#### 2.4.2.1 Wide & Deep Learning for Recommender Systems

Mô hình tuyến tính tổng quát với các phép biến đổi đặc trưng phi tuyến được sử dụng rộng rãi cho các bài toán phân loại và hồi quy quy mô lớn với đầu vào thưa thớt. Việc ghi nhớ các tương tác tính năng thông qua một loạt các chuyển đổi tính năng sản phẩm chéo có hiệu quả và có thể diễn giải được, trong khi việc tổng quát hóa đòi hỏi nhiều nỗ lực về kỹ thuật tính năng hơn. Với kỹ thuật ít tính năng hơn, mạng nơ-ron sâu có thể tổng quát hóa tốt hơn cho các tổ hợp tính năng không nhìn thấy thông qua các phép nhúng dày đặc chiều thấp đã học cho các đối tượng địa lý thưa thớt. Tuy nhiên, mạng nơ-ron sâu có tính năng nhúng có thể tổng quát hóa quá mức và đề xuất các mục ít liên quan hơn khi các tương tác giữa người dùng với mục này thưa thớt và có thứ hạng cao. Trong bài báo này, chúng tôi trình bày Học tập rộng và sâu — các mô hình tuyến tính rộng và mạng nơ-ron sâu được đào tạo chung — để kết hợp các lợi ích của việc ghi nhớ và khái quát hóa cho các hệ thống khuyến nghị. Chúng tôi đã sản xuất và đánh giá hệ thống trên Google Play, một cửa hàng ứng dụng dành cho thiết bị di động thương mại với hơn một tỷ người dùng đang hoạt động và hơn một triệu ứng dụng. Kết quả thử nghiệm trực tuyến cho thấy rằng Wide & Deep đã tăng đáng kể lượt chuyển đổi ứng dụng so với các mô hình chỉ dành cho phạm vi rộng và chỉ sâu. Chúng tôi cũng đã triển khai nguồn mở của mình trong TensorFlow.



##### 2.4.2.1.1 Thành phần rộng

Thành phần rộng là một mô hình tuyến tính tổng quát có dạng y = wT x + b, y là dự đoán, x = [x1, x2, ..., xd] là vectơ của d đối tượng, w = [w1, w2, ..., wd] là tham số mô hình và b là độ lệch. Bộ tính năng bao gồm các tính năng đầu vào thô và các tính năng đã biến đổi. Một trong những chuyển đổi quan trọng nhất là chuyển đổi sản phẩm chéo, được định nghĩa là:



trong đó c\_ki là một biến boolean là 1 nếu đặc trưng thứ i là một phần của phép biến đổi thứ k φk, và 0 nếu ngược lại. Đối với các đối tượng địa lý nhị phân, chuyển đổi sản phẩm chéo (ví dụ: “VÀ (giới tính = nữ, ngôn ngữ = vi)”) là 1 nếu và chỉ khi các đối tượng địa lý cấu thành (“giới tính = nữ” và “ngôn ngữ = vi”) đều là 1 , và 0 nếu không. Điều này nắm bắt các tương tác giữa các đối tượng nhị phân và thêm tính phi tuyến tính vào mô hình tuyến tính tổng quát.

##### 2.4.2.1.2 Thành phần sâu

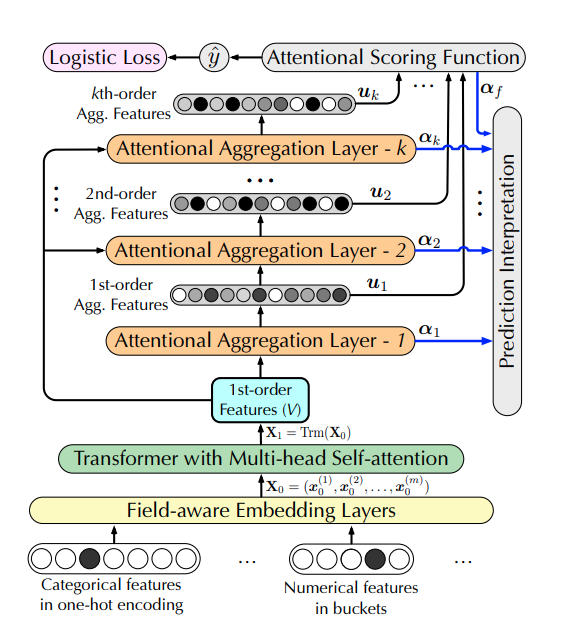
Thành phần sâu là một mạng nơ-ron truyền tới, như thể hiện trong Hình 1 (bên phải). Đối với các đối tượng địa lý phân loại, đầu vào ban đầu là các chuỗi đối tượng địa lý (ví dụ: “language = en”). Mỗi đối tượng phân loại có chiều cao, thưa thớt này lần đầu tiên được chuyển đổi thành một vectơ có giá trị thực có chiều thấp và dày đặc, thường được gọi là vectơ nhúng. Kích thước của các lần nhúng thường theo thứ tự từ O (10) đến O (100). Các vectơ nhúng được khởi tạo ngẫu nhiên và sau đó các giá trị được huấn luyện để giảm thiểu hàm mất mát cuối cùng trong quá trình huấn luyện mô hình. Sau đó, các vectơ nhúng dày đặc chiều thấp này được đưa vào các lớp ẩn của mạng nơ-ron trong chuyển tiếp. Cụ thể, mỗi lớp ẩn thực hiện tính toán sau:



trong đó l là số lớp và f là hàm kích hoạt, thường là các đơn vị tuyến tính được chỉnh lưu (ReLU). a (l), b (l), và W (l) là kích hoạt, độ chệch và trọng lượng mô hình ở lớp thứ l.

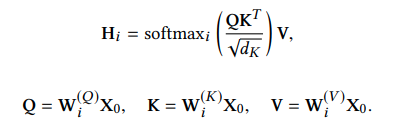
#### 2.4.2.2 Interpretable Click-Through Rate Prediction through Hierarchical Attention

Dự đoán tỷ lệ nhấp (CTR) là một nhiệm vụ quan trọng trong quảng cáo và tiếp thị trực tuyến. Đối với vấn đề này, các phương pháp tiếp cận hiện tại, với kiến ​​trúc nông hoặc sâu, đều có ba nhược điểm lớn. Đầu tiên, họ thường thiếu các lý do thuyết phục để giải thích kết quả của các mô hình. Các dự đoán và khuyến nghị không thể giải thích được có thể khó xác thực và do đó không đáng tin cậy và không đáng tin cậy. Trong nhiều ứng dụng, những đề xuất không phù hợp thậm chí có thể mang lại hậu quả nặng nề. Thứ hai, các cách tiếp cận hiện tại có hiệu quả kém trong việc phân tích các tương tác tính năng bậc cao. Thứ ba, tính đa nghĩa của các tương tác đặc trưng trong các không gian con ngữ nghĩa khác nhau phần lớn bị bỏ qua. Trong bài báo này, chúng tôi đề xuất InterHAt sử dụng transformer có cơ chế attention để học tính năng. Trên hết, các lớp attention phân cấp được sử dụng để dự đoán CTR đồng thời cung cấp thông tin chi tiết có thể giải thích được về kết quả dự đoán. InterHAt nắm bắt các tương tác tính năng bậc cao bằng một chiến lược tổng hợp có attention hiệu quả với độ phức tạp tính toán thấp. Các thí nghiệm mở rộng trên bốn tập dữ liệu thực công khai và một tập dữ liệu tổng hợp chứng minh hiệu lực và hiệu quả của InterHAt.



##### 2.4.2.2.1 Multi-head Transformer

Transformer phổ biến trong NLP nhờ sức mạnh vượt trội để học các đồng tác động đến ngữ nghĩa văn bản của các cặp từ trong một câu hoặc giữa các câu bất kể thứ tự và khoảng cách của các từ. Trong ngữ cảnh dự đoán CTR, chúng tôi xác định hệ số của các đối tượng địa lý, tức là các tương tác đối tượng địa lý, theo hướng phân cực khác nhau như là "đa nghĩa". Do đó, chúng tôi trang bị cho InterHAt một Transformer dựa trên sự attention nhiều đầu vào để nắm bắt các tương tác tính năng theo cặp phong phú và tìm hiểu sự đa dạng của các tương tác tính năng trong các không gian con ngữ nghĩa khác nhau, tức là các hàm ý đa dạng đối với CTR trong các ngữ cảnh nhấp qua khác nhau . Với ma trận đầu vào X0 có chứa các tính năng nhúng có thể học được của một bản ghi CTR huấn luyện, biểu diễn tiềm ẩn Hi của đầu transformer i thu được bằng sự attention đến sản phẩm theo tỷ lệ.



Sự kết hợp của các tính năng ẩn Hi tạo thành một ma trận biểu diễn tăng cường X1 để lưu giữ cả thông tin nội tại và đa giác của mỗi đối tượng địa lý. Về mặt tính toán, chúng tôi sử dụng kết hợp theo sau là lớp chuyển tiếp và ReLU cho kết hợp để tìm hiểu tính phi tuyến tính của thông tin được kết hợp như



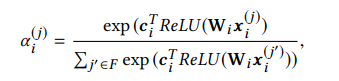
trong đó Wm ∈ R(d × hdk) chứa các trọng số và h là số đầu attention và “;” biểu thị sự nối các ma trận. X1 ∈ R(d × m) là ma trận có các tính năng đa dạng được tăng cường và sẵn sàng được gửi đến lớp attention phân cấp để dự đoán CTR có thể giải thích được.

##### 2.4.2.2 Hierarchical Attention

Ma trận tính năng tăng cường X1 được phục vụ như là đầu vào của các lớp attention phân cấp học sự tương tác của tính năng và tạo ra các diễn giải đồng thời. Tuy nhiên, việc tính toán các tương tác đa tính năng bậc cao bằng cách liệt kê tất cả các tổ hợp có thể có là tốn kém do sự bùng nổ tổ hợp. Chi phí tiềm năng như vậy thúc đẩy việc tổng hợp đơn đặt hàng hiện tại trước khi tiếp tục tính toán đơn đặt hàng cao hơn. Nghĩa là, để tạo ra các tính năng chéo thứ (i +1) Xi + 1, trước tiên chúng tôi tổng hợp các tính năng ẩn của lớp thứ i thành ui như một bản tóm tắt của Xi. Tương tác giữa Xi và X1, từ đó chúng ta suy ra Xi + 1, được tính bằng đại lượng của Xi, tức là, tập hợp có attention ui từ Phương trình (1) và X1. Về mặt toán học, cho trước ma trận đặc trưng thứ i Xi = x (1) i ,. . . , x (m) i, biểu diễn tổng hợp có attention của nó u\_i là



trong đó α (j) i ∈ R biểu thị sự attention trên trường thứ j trong lớp tập hợp attention thứ i. α (j) i được tính bởi:



trong đó Wi ∈ R s × d là trọng lượng của lớp i, ci ∈ R s là vectơ ngữ cảnh của lớp i và s biểu thị kích thước không gian attention. Lưu ý rằng các cơ chế attention khác cũng có thể được áp dụng ở đây, chẳng hạn như cơ chế attention theo dõi. Sử dụng ui và Xi, chúng ta suy ra x (j) i + 1 trong Xi + 1 bằng phép biến đổi tích chéo.



trong đó ◦ biểu thị tích Hadamard của hai vectơ.

#### 2.4.2.3 Evaluating collaborative filtering recommender systems

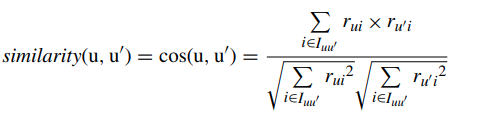
Do sự bùng nổ của thông tin có sẵn trên Internet, nhu cầu về các phương tiện hữu hiệu để truy cập và xử lý chúng đã trở nên quan trọng đối với tất cả mọi người. Hệ thống đề xuất đã được phát triển để giúp người dùng tìm thấy những gì họ có thể quan tâm và chủ doanh nghiệp bán sản phẩm của họ hiệu quả hơn. Họ đã nhận thấy nhiều sự attention trong cả lĩnh vực học thuật và công nghiệp. Thuật toán giới thiệu tính đến các tương tác giữa người dùng với mặt hàng, tức là lịch sử xếp hạng (hoặc mua hàng) của người dùng trên các mặt hàng và thông tin ngữ cảnh của họ, nếu có. Sau đó, nó cung cấp danh sách các mặt hàng tiềm năng cho từng người dùng mục tiêu, sao cho người dùng có khả năng đánh giá tích cực (hoặc mua) chúng. Trong bài báo này, chúng tôi xem xét các chỉ số đánh giá được sử dụng để đánh giá hiệu suất của các thuật toán đề xuất. Chúng tôi cũng khảo sát một số thuật toán đề xuất cổ điển và hiện đại và so sánh hiệu suất của chúng theo các chỉ số đánh giá khác nhau trên năm tập dữ liệu điểm chuẩn. Các thử nghiệm của chúng tôi cho thấy rằng không có thuật toán đề xuất vàng nào cho thấy hiệu suất tốt nhất trong tất cả các chỉ số đánh giá. Chúng tôi cũng tìm thấy sự khác biệt lớn trên các tập dữ liệu. Điều này chỉ ra rằng người ta nên xem xét cẩn thận các tiêu chí đánh giá trong việc lựa chọn thuật toán khuyến nghị cho một ứng dụng cụ thể.

Các RS của Lọc cộng tác (CF) xem xét các điểm tương đồng về mặt hàng hoặc dựa trên người dùng và trích xuất danh sách các đề xuất dựa trên chúng. Để chính thức hóa điều này, hãy biểu thị '' U '' là tập hợp người dùng trong hệ thống, được hiển thị bởi U = {u1, u2, ..., um} và '' I '' là tập hợp các mục, được đại diện bởi I = {I1, I2, ..., In}. rui đại diện cho xếp hạng do người dùng u đưa ra cho mục i và thường có thang số, chẳng hạn như xếp hạng năm sao cho Movielens (1 có nghĩa là rất tệ và 5 rất tốt) và thang xếp hạng mười sao cho IMDB.2 Trong một số trường hợp, thang đo là liên tục, ví dụ, rui nằm trong khoảng [−10, 10] đối với người khuyến nghị Jester Joke. Có những cách rõ ràng hoặc ẩn ý để thu thập xếp hạng của người dùng, như một chỉ báo về sở thích của họ đối với các mặt hàng. Xếp hạng số dọc theo nhị phân (thích / không thích) và đơn phân (ví dụ: lựa chọn Thích duy nhất trong Facebook), được nhập trực tiếp bởi người dùng, được coi là xếp hạng rõ ràng. Tệp dữ liệu chứa xếp hạng (được gọi là ma trận tương tác giữa người dùng với mặt hàng) được sử dụng để tìm hiểu sở thích và thói quen của người dùng, dự đoán xếp hạng mới, đề xuất mặt hàng cho người dùng và cuối cùng là đánh giá hệ thống. Đối với mục đích đánh giá, người ta cần chia ma trận đánh giá thành hai phần: tập huấn luyện để học tập và tập kiểm tra để đánh giá kết quả hoạt động. Các thuật toán CF dựa trên bộ nhớ và dựa trên mô hình sử dụng các ma trận này theo cách khác nhau, được thảo luận trong các phần sau.

##### 2.4.2.3.1 MEMORY-BASED COLLABORATIVE FILTERING

Như đã đề cập, các thuật toán dựa trên bộ nhớ dự đoán xếp hạng mới dựa trên dữ liệu có sẵn (được tải vào bộ nhớ), sử dụng sự tương đồng của người dùng hoặc mục khác với người dùng / mục đích. Tập hợp những người dùng tương tự với người dùng mục tiêu (hoặc các mục tương tự với một mục đích) được gọi là nhóm hàng xóm của họ và được sử dụng để trích xuất những người dùng / mục có lịch sử xếp hạng tương tự. Giả định cơ bản là nếu hai người dùng có lịch sử xếp hạng giống nhau cho các mặt hàng chung, họ có thể sẽ có các sở thích tương tự đối với các mặt hàng còn lại. Đối với hai mục được một số người dùng đánh giá tương tự nhau, chúng có thể sẽ được những người dùng còn lại đánh giá theo cùng một cách. Tuy nhiên, luôn có những cá nhân có sở thích và sở thích riêng biệt sẽ không giúp ích gì cho trường hợp này, nhưng nói chung, giả định này đã tỏ ra hữu ích. Sau khi hình thành vùng lân cận, xếp hạng mới cho cặp mặt hàng người dùng mục tiêu được dự đoán dưới dạng hàm số xếp hạng của những người hàng xóm cho mặt hàng cụ thể đó và mức độ tương đồng của chúng với người dùng mục tiêu. Dựa trên việc sử dụng người dùng mục tiêu / hàng xóm, các thuật toán này được chia thành hai loại CF dựa trên người dùng và dựa trên mặt hàng.

Có nhiều biện pháp tương tự khác nhau để trích xuất tập hợp các lân cận, chẳng hạn như độ tương tự dựa trên cosin được điều chỉnh từ truy xuất thông tin, cosin đã điều chỉnh, hệ số tương quan Pearson, tương quan Pearson bị ràng buộc, Euclidean và chênh lệch bình phương trung bình. Ngoài sự giống nhau, trong một số trường hợp, người ta cũng có thể sử dụng các giá trị khác biệt, ví dụ, khi độ thưa thớt của dữ liệu có sẵn cao và mức độ liên quan trở nên quan trọng hơn mối tương quan. Tiếp theo, chúng tôi xem xét sự tương tự dựa trên cosine và tương quan Pearson; tuy nhiên vì hệ số tương quan Pearson đã cho thấy là kỹ thuật hiệu quả hơn, chúng tôi đã chọn Pearson làm thước đo độ tương đồng trong việc triển khai. Sự tương tự dựa trên cosine giữa hai người dùng u và u\_0 được tính bằng cách sử dụng Eq. như sau :



##### 2.4.2.3.2 MODEL-BASED COLLABORATIVE FILTERING

Các thuật toán CF dựa trên mô hình sử dụng các kỹ thuật khác nhau trên tập huấn luyện, để tìm các mẫu trong dữ liệu và học một mô hình để dự đoán xếp hạng mới. Người ta có thể đặt tên Slope một, các mô hình nhân tố tiềm ẩn như Ma trận thừa số (MF) và Phân tích giá trị số ít (SVD), bộ phân loại Bayes, mô hình phân cụm, các mô hình quan hệ xác suất khác nhau, phân tích ngữ nghĩa tiềm ẩn xác suất, hồi quy tuyến tính , mô hình entropy cực đại, Phân bổ Dirichlet tiềm ẩn, Mô hình dựa trên chuỗi Markov, phân tích thành phần chính (PCA), phân tích nhân tố xác suất, mạng nơ-ron và hệ thống mờ trong số các kỹ thuật được sử dụng cho CF dựa trên mô hình. Sau đây, chúng tôi xem xét một số phương pháp dựa trên mô hình thường được sử dụng. Mặc dù có một tài liệu phong phú về CF dựa trên mô hình, các thuật toán này có khả năng ứng dụng hạn chế trong các tình huống thực tế.

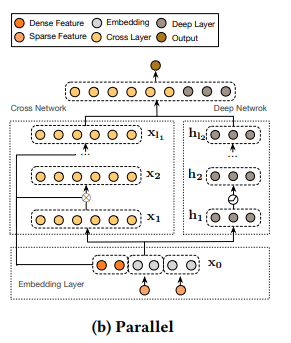
### 2.4.3 Cấu trúc song song và cấu trúc xếp chồng

#### 2.4.3.1 Cấu trúc song song

Một dòng công việc cùng đào tạo hai mạng song song lấy cảm hứng từ mô hình rộng và sâu, trong đó thành phần rộng lấy đầu vào là các điểm giao nhau của các đặc trưng thô; và thành phần sâu là một mô hình DNN. Tuy nhiên, việc chọn các đối tượng địa lý chéo cho thành phần rộng trở lại vấn đề kỹ thuật tính năng đối với mô hình tuyến tính. Tuy nhiên, mô hình rộng và sâu đã truyền cảm hứng cho nhiều công trình áp dụng kiến ​​trúc song song này và cải thiện thành phần rộng.

Mạng nơ-ron parareal bao gồm các cấu trúc tốt có thể được xử lý song song và một cấu trúc thô xấp xỉ các cấu trúc tốt bằng cách mô phỏng một trong các thuật toán song song trong thời gian được gọi là parareal. Không giống như các phương pháp hiện có được đề cập ở trên, mạng thần kinh thực có thể giảm đáng kể thời gian giao tiếp giữa các GPU vì các cấu trúc nhỏ không giao tiếp với nhau mà chỉ giao tiếp với cấu trúc thô. Do đó, phương pháp được đề xuất có hiệu quả trong việc giảm thời gian trôi qua để xử lý các mạng nơ-ron rất sâu. Kết quả số xác nhận rằng mạng thần kinh thực cung cấp hiệu suất tương tự hoặc tốt hơn so với mạng gốc ngay cả với thời gian đào tạo ít hơn.

Mạng thần kinh thực. Trong phần này, chúng tôi đề xuất một phương pháp để thiết kế một mạng nơ-ron parareal bằng cách mô phỏng thuật toán parareal được giới thiệu trong Phần 2 từ một mạng nơ-ron nguồn cấp tới cho trước. Mạng neural parareal thu được có cấu trúc song song nội tại và thích hợp cho tính toán song song bằng cách sử dụng nhiều GPU với bộ nhớ phân tán đồng thời.



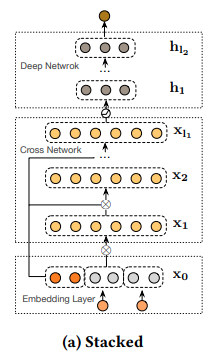
#### 2.4.3.2 Cấu trúc xếp chồng

Một dòng công việc khác giới thiệu một lớp tương tác — lớp này tạo ra các điểm giao nhau rõ ràng — giữa lớp nhúng và mô hình DNN. Lớp tương tác này nắm bắt tương tác của tính năng ở giai đoạn đầu và tạo điều kiện thuận lợi cho việc học các lớp ẩn tiếp theo. Mạng nơ-ron dựa trên sản phẩm (PNN) giới thiệu lớp sản phẩm bên trong (IPNN) và bên ngoài (OPNN) là các lớp tương tác theo cặp. Một nhược điểm của OPNN nằm ở chi phí tính toán cao. Neural FM (NFM) mở rộng FM bằng cách thay thế sản phẩm bên trong bằng sản phẩm Hadamard; DLRM theo sau FM để tính toán tính năng đi qua các sản phẩm bên trong; Các mô hình này chỉ có thể tạo tối đa các dấu thập rõ ràng bậc 2. AFN chuyển đổi các đối tượng địa lý thành không gian nhật ký và học hỏi một cách thích nghi các tương tác đối tượng đặt hàng tùy ý. Tương tự như DeepFM và xDeepFM, chúng chỉ chấp nhận các nhúng có kích thước bằng nhau.

Mạng nơron sâu dựa trên xếp chồng (S-DNN) được tổng hợp với đa nguyên của các mô-đun học tập cơ bản, nối tiếp nhau, để tổng hợp một mạng nơ-ron sâu (DNN) thay thế cho việc phân loại mẫu. Trái ngược với các DNN được đào tạo từ đầu đến cuối bằng cách nhân giống ngược (BP), mỗi lớp S-DNN, tức là một mô-đun có thể tự phân bổ, phải được đào tạo một cách dứt khoát và độc lập mà không có sự can thiệp của BP. Trong bài báo này, một S-DNN dựa trên hồi quy sườn núi, được gọi là mạng phân tích sâu (DAN), cùng với sự kết hợp của nó (K-DAN), được thiết kế để tái phân tích tính năng nhiều lớp từ các tính năng cơ sở được trích xuất trước và các tính năng có cấu trúc. Công thức lý thuyết của chúng tôi chứng minh rằng DAN / KDAN học lại bằng cách xáo trộn các biến thể nội bộ / giữa các lớp, ngoài việc giảm các lỗi dự đoán. Chúng tôi xem xét kỹ lưỡng hiệu suất DAN / KDAN để phân loại mẫu trên tập dữ liệu của các miền khác nhau — khuôn mặt, chữ số viết tay, đối tượng chung chung, cho đến tên một số. Không giống như các DNN được tối ưu hóa BP điển hình được đào tạo từ bộ dữ liệu khổng lồ bằng GPU, chúng tôi tiết lộ rằng DAN / K-DAN chỉ có thể đào tạo bằng cách sử dụng CPU ngay cả đối với các bộ đào tạo quy mô nhỏ. Kết quả thử nghiệm của chúng tôi cho thấy rằng DAN / K-DAN hoạt động tốt hơn các S-DNN hiện tại và cả các DNN được BP đào tạo, bao gồm perceptron nhiều người chơi, mạng niềm tin sâu, v.v. mà không áp dụng tăng cường dữ liệu.

Cấu trúc DAN được xếp chồng lên nhau với các đơn vị đã học để tái phân tích tính năng theo lớp. Theo các định nghĩa trong Phần II, giả sử N ≥ d, lớp đầu tiên của DAN có độ sâu L được giao với X, và được mở rộng để ước lượng tập trọng số giải tích W∈ Rd × Nc cho = 1, ..., L như sau:





### 2.4.4 Kiến trúc mô hình

#### 2.4.4.1 Embedding Layer

Lớp nhúng nhận đầu vào là sự kết hợp của các đặc trưng phân loại (thưa thớt) và dày đặc, và đầu ra x0 ∈ R 𝑑. Đối với đặc trưng phân loại thứ, chúng tôi chiếu nó từ không gian thưa chiều cao sang không gian dày đặc chiều thấp hơn thông qua xembed, 𝑖 = 𝑊embed, 𝑖e𝑖, trong đó e𝑖 ∈ {0, 1} 𝑣𝑖; 𝑊 ∈ R 𝑒𝑖 × 𝑣𝑖 là một ma trận chiếu đã học; xembed, 𝑖 ∈ R 𝑒𝑖 là vectơ nhúng dày đặc; 𝑣𝑖 và 𝑒𝑖 đại diện cho kích thước vocab và kích thước nhúng tương ứng. Đối với các đối tượng đa giá trị, chúng tôi sử dụng giá trị trung bình của các vectơ nhúng làm vectơ cuối cùng.

Đầu ra là sự kết hợp của tất cả các vectơ được nhúng và các đặc trưng dày đặc chuẩn hóa: x0 = [x\_embed, 1; . . . ; x\_embed, 𝑛; 𝑥\_dense].

Không giống như nhiều tác phẩm có liên quan [13, 16, 26, 34, 35, 46] yêu cầu 𝑒𝑖 = 𝑒𝑗 ∀𝑖, 𝑗, mô hình của chúng tôi chấp nhận kích thước nhúng tùy ý. Điều này đặc biệt quan trọng đối với các khuyên công nghiệp khi kích thước vocab thay đổi từ 𝑂 (10) đến 𝑂 (105). Hơn nữa, mô hình của chúng tôi không giới hạn ở phương pháp nhúng được mô tả ở trên; có thể áp dụng bất kỳ kỹ thuật nhúng nào khác như băm.

Chúng tôi coi dữ liệu đầu vào có đặc điểm thưa thớt và dày đặc. Trong các hệ thống đề xuất trên trang web như dự đoán CTR, đầu vào chủ yếu là các tính năng phân loại, ví dụ: "country = usa". Các tính năng như vậy được mã hóa dưới dạng vectơ một nóng, ví dụ: "[0,1,0]"; tuy nhiên, điều này dẫn đến không gian đặc trưng có chiều cao quá mức cho các từ vựng lớn.

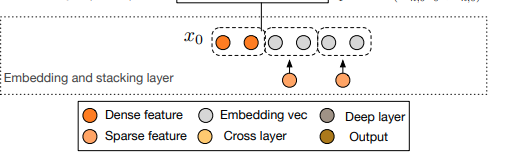
Để giảm kích thước, chúng tôi sử dụng một thủ tục nhúng để biến đổi các đối tượng nhị phân này thành các vectơ dày đặc của giá trị thực (thường được gọi là vectơ nhúng):



trong đó xembed, i là vector nhúng, xi là đầu vào nhị phân trong danh mục thứ i và Wembed, i ∈ R ne × nv là ma trận nhúng tương ứng sẽ được tối ưu hóa cùng với các tham số khác trong mạng và ne, nv lần lượt là kích thước nhúng và kích thước từ vựng.

Cuối cùng, chúng tôi xếp các vectơ nhúng, cùng với các đối tượng địa lý dày đặc được chuẩn hóa x\_dense, thành một vectơ:





#### 2.4.4.2 Cross Network

Cốt lõi của DCN-V2 nằm ở các lớp chéo tạo ra các điểm giao nhau về tính năng rõ ràng. Phương trình (1) hiển thị lớp chéo thứ (𝑙 + 1).



trong đó x0 ∈ R 𝑑 là lớp cơ sở chứa các tính năng ban đầu của bậc 1 và thường được đặt làm lớp nhúng (đầu vào). x𝑙, x𝑙 + 1 ∈ R 𝑑, tương ứng, đại diện cho đầu vào và đầu ra của lớp chéo thứ (𝑙 + 1). 𝑊𝑙 ∈ R 𝑑 × 𝑑 và b𝑙 ∈ R 𝑑 là ma trận trọng số và véc tơ thiên vị đã học. Hình 2 cho thấy cách một lớp chéo riêng lẻ hoạt động.

Đối với mạng chéo nhiều lớp, bậc đa thức cao nhất là 𝑙 + 1 và mạng chứa tất cả các điểm chéo của đối tượng cho đến bậc cao nhất. Vui lòng xem Phần 4.1 để biết phân tích chi tiết, cả từ quan điểm bitwise và đặc điểm. Khi 𝑊 = 1 × w⊤, trong đó 1 đại diện cho một vectơ của chúng, thì DCN-V2 rơi trở lại DCN.

Các lớp chéo chỉ có thể tái tạo các lớp hàm đa thức có mức độ giới hạn; bất kỳ không gian hàm phức tạp nào khác chỉ có thể là gần đúng. Do đó, chúng tôi giới thiệu một mạng sâu bên cạnh để bổ sung cho việc mô hình hóa phân phối vốn có trong dữ liệu.

Tương tác mức độ cao trên các tính năng. Cấu trúc đặc biệt của mạng lưới chéo làm cho mức độ của các tính năng chéo tăng lên theo độ sâu của lớp. Bậc đa thức cao nhất (xét theo đầu vào x0) đối với mạng chéo lớp l là +1. Trên thực tế, mạng chéo bao gồm tất cả các số hạng chéo x α1 1 x α2 2. . . x αd d có li độ từ 1 đến l + 1.

Độ phức tạp về thời gian và không gian của một mạng chéo là tuyến tính trong kích thước đầu vào. Trước đó, một mạng chéo tạo ra độ phức tạp không đáng kể so với đối tác sâu của nó, giữ cho độ phức tạp tổng thể của DCN ở cùng mức độ của một DNN truyền thống. Tính hiệu quả Œlà lợi ích từ thuộc tính hạng một của x0x T l, cho phép chúng tôi tạo ra tất cả các số hạng chéo mà không cần tính toán hoặc lưu trữ toàn bộ ma trận.

Một số lượng nhỏ các tham số của mạng chéo đã hạn chế dung lượng của mô hình. Để nắm bắt các tương tác phi tuyến tính cao, chúng tôi giới thiệu một mạng sâu song song.

#### 2.4.4.3 Deep Network

Mạng sâu là một mạng nơ-ron chuyển tiếp được kết nối đầy đủ, với mỗi lớp sâu có công thức sau:

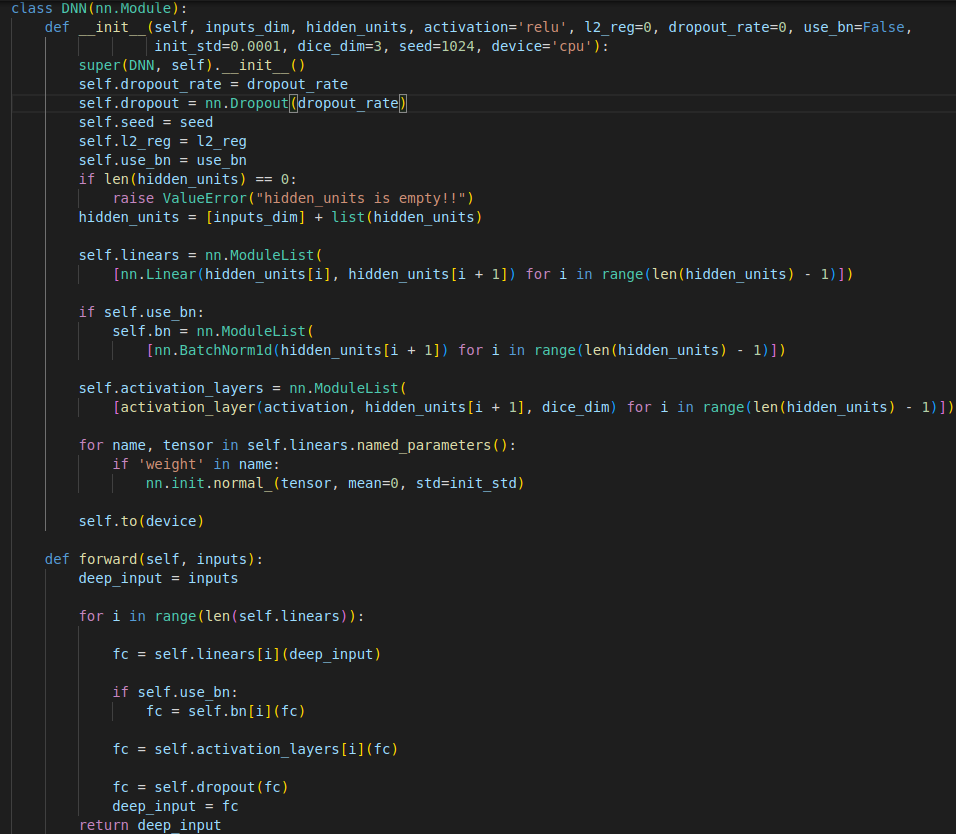


trong đó hl ∈ R nl, hl + 1 ∈ R nl + 1 lần lượt là lớp ẩn thứ l và (l + 1) -th; Wl ∈ R nl + 1 × nl, bl ∈ R nl + 1 là các tham số cho lớp sâu thứ l; và f (·) là hàm ReLU.

Phân tích độ phức tạp. Để đơn giản, chúng tôi giả sử tất cả các lớp sâu có kích thước bằng nhau. Gọi Ld là số lớp sâu và m là kích thước lớp sâu. Œen, số lượng tham số trong mạng sâu là:



Mã nguồn:



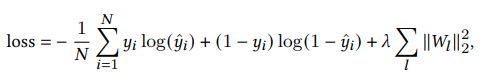
#### 2.4.4.4 Deep and Cross Combination

Chúng tôi tìm kiếm các cấu trúc để kết hợp mạng chéo và mạng sâu. Văn học gần đây áp dụng hai cấu trúc: xếp chồng lên nhau và song song. Trong thực tế, chúng tôi nhận thấy rằng kiến ​​trúc nào hoạt động tốt hơn là phụ thuộc vào dữ liệu. Do đó, chúng tôi trình bày cả hai cấu trúc:

Cấu trúc xếp chồng: Đầu vào x0 được cấp cho mạng chéo, theo sau là mạng sâu và lớp cuối cùng được đưa ra bởi xfinal = h𝐿𝑑, h0 = x𝐿𝑐, mô hình hóa dữ liệu dưới dạng 𝑓deep ◦ 𝑓cross.

Cấu trúc song song: Đầu vào x0 được cấp song song cho cả mạng chéo và mạng sâu; sau đó, các đầu ra x𝐿𝑐 và h𝐿𝑑 được nối với nhau để tạo ra lớp đầu ra cuối cùng x\_final = [x𝐿𝑐; h𝐿𝑑]. Cấu trúc này lập mô hình dữ liệu dưới dạng 𝑓cross + 𝑓deep.

Cuối cùng, dự đoán 𝑦 ^ 𝑖 được tính là: 𝑦 ^ 𝑖 = 𝜎 (w⊤ logitxfinal), trong đó wlogit là vectơ trọng số của logit và 𝜎 (𝑥) = 1 / (1 + exp (−𝑥) ). Đối với tổn thất cuối cùng, chúng tôi sử dụng Log Loss thường được sử dụng để học xếp hạng các hệ thống, đặc biệt là với nhãn nhị phân (ví dụ: nhấp chuột). Lưu ý rằng bản thân DCN-V2 vừa là dự đoán-nhiệm vụ vừa là bất khả tri về mất chức năng.

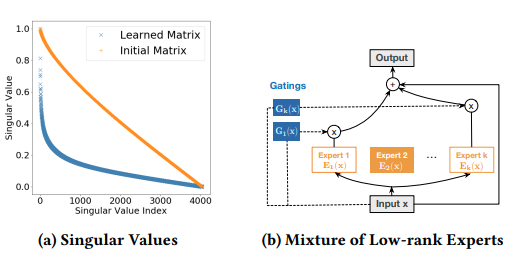


trong đó 𝑦 ^ 𝑖 ’s là các dự đoán; 𝑦𝑖 là các nhãn thực sự; 𝑁 là tổng số đầu vào; và 𝜆 là tham số chính quy 𝐿2.

#### 2.4.4.5 Cost-Effective Mixture of Low-Rank DCN

Trong các mô hình sản xuất thực, năng lực của mô hình thường bị hạn chế bởi các nguồn lực phục vụ hạn chế và các yêu cầu nghiêm ngặt về độ trễ. Thông thường, chúng ta phải tìm kiếm các phương pháp để giảm chi phí mà vẫn duy trì độ chính xác. Các kỹ thuật hạng thấp [12] được sử dụng rộng rãi [5, 9, 14, 20, 51, 52] để giảm chi phí tính toán. Nó xấp xỉ một ma trận đặc 𝑀 ∈ R 𝑑 × 𝑑 bởi hai ma trận cao và gầy 𝑈, 𝑉 ∈ R 𝑑 × 𝑟. Khi 𝑟 ≤ 𝑑 / 2, chi phí sẽ giảm xuống. Tuy nhiên, chúng có hiệu quả nhất khi ma trận cho thấy khoảng cách lớn về các giá trị kỳ dị hoặc phân rã phổ nhanh. Trong nhiều cài đặt, chúng tôi thực sự nhận thấy rằng ma trận đã học được xếp hạng thấp về mặt số học trong thực tế.

Hình cho thấy dạng phân rã kỳ dị của ma trận đã học 𝑊 trong DCN-V2 (xem Phương trình (1)) từ một mô hình sản xuất. So với ma trận ban đầu, ma trận đã học cho thấy dạng phân rã phổ nhanh hơn nhiều. Hãy xác định thứ hạng số 𝑅𝑇 với dung sai T là argmin𝑘 (𝜎𝑘 <𝑇 · 𝜎1), trong đó 𝜎1 ≥ 𝜎2 ≥,. . . , ≥ 𝜎𝑛 là các giá trị đơn lẻ. Khi đó, 𝑅𝑇 có nghĩa là phần lớn khối lượng đến dung sai 𝑇, được bảo toàn trong các giá trị số ít nhất. Trong lĩnh vực máy học và học sâu, một mô hình vẫn có thể hoạt động tốt một cách đáng ngạc nhiên với dung sai cao hợp lý 𝑇 2.

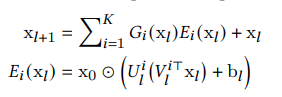


Do đó, việc áp đặt cấu trúc cấp thấp cho 𝑊 là rất có động cơ. Phương trình (2) hiển thị kết quả (𝑙 + 1) lớp chéo thứ hạng thấp



trong đó 𝑈𝑙, 𝑉𝑙 ∈ R 𝑑 × 𝑟 và 𝑟 ≪ 𝑑. Phương trình (2) có hai cách giải thích: 1) chúng ta tìm hiểu các giao nhau của đối tượng trong một không gian con; 2) chúng tôi chiếu đầu vào x thành R 𝑟 chiều thấp hơn, và sau đó chiếu nó trở lại R 𝑑. Hai cách giải thích đã truyền cảm hứng cho hai cải tiến mô hình sau

Diễn giải 1 truyền cảm hứng cho chúng tôi áp dụng ý tưởng từ Mixture-ofExperts (MoE) [10, 19, 30, 45]. Mô hình dựa trên MoE bao gồm hai thành phần: chuyên gia (thường là một mạng nhỏ) và gating (một chức năng của đầu vào). Trong trường hợp của chúng tôi, thay vì dựa vào một chuyên gia duy nhất (Phương trình (2)) để tìm hiểu các điểm kết hợp tính năng, chúng tôi sử dụng nhiều chuyên gia như vậy, mỗi tính năng học tập tương tác trong một không gian con khác nhau và kết hợp một cách thích ứng các điểm kết hợp đã học bằng cách sử dụng cơ chế kiểm tra phụ thuộc vào đầu vào x. Hỗn hợp kết quả của công thức lớp chéo cấp thấp được thể hiện trong Eq. (3).



trong đó 𝐾 là số lượng chuyên gia; 𝐺𝑖 (·): R 𝑑 ↦ → R là hàm gating, sigmoid chung hoặc softmax; 𝐸𝑖 (·): R 𝑑 ↦ → R 𝑑 là chuyên gia thứ nhất trong việc học các tính năng lai. 𝐺 (·) tự động trọng số từng chuyên gia cho đầu vào x và khi 𝐺 (·) ≡ 1, phương trình (3) giảm trở lại phương trình (2).

Sự diễn giải 2 truyền cảm hứng cho chúng tôi để tận dụng tính chất chiều thấp của không gian được chiếu. Thay vì chiếu ngược ngay lập tức từ thứ nguyên 𝑑 ′ sang 𝑑 (𝑑 ′ ≪ 𝑑), chúng tôi áp dụng thêm các phép biến đổi phi tuyến trong không gian chiếu để tinh chỉnh biểu diễn.



trong đó 𝑔 (·) đại diện cho bất kỳ chức năng kích hoạt phi tuyến nào.

Các cuộc thảo luận. Phần này nhằm mục đích sử dụng hiệu quả ngân sách thời gian / bộ nhớ cố định để tìm hiểu các tính năng có ý nghĩa. Từ phương trình (1) - (4), mỗi công thức đại diện cho một lớp hàm lớn hơn hoàn toàn giả sử #params cố định.

Khác với nhiều kỹ thuật nén mô hình trong đó quá trình nén được thực hiện sau quá trình đào tạo, mô hình của chúng tôi áp đặt cấu trúc trước khi đào tạo và cùng tìm hiểu các tham số liên quan với phần còn lại của các tham số. Do đó, lớp chéo là một phần tích hợp của hệ phi tuyến 𝑓 (x) = 𝑓𝑘 (𝑊𝑘) ◦ · · · ◦ 𝑓1 (𝑊1) (x), trong đó (𝑓𝑖 + 1 ◦ 𝑓𝑖) (·) B 𝑓𝑖 + 1 (𝑓𝑖 (·)). Do đó, động lực đào tạo của hệ thống tổng thể có thể bị ảnh hưởng và sẽ rất thú vị khi xem các thống kê toàn cầu, chẳng hạn như ma trận Jacobian và Hession của 𝑓 (x), bị ảnh hưởng như thế nào. Chúng tôi để những cuộc điều tra như vậy cho công việc trong tương lai

#### 2.4.4.6 Complexity Analysis

Gọi 𝑑 biểu thị kích thước nhúng, 𝐿𝑐 biểu thị số lớp chéo, 𝐾 biểu thị số lượng chuyên gia DCN cấp thấp. Hơn nữa, để đơn giản, chúng tôi giả sử mỗi chuyên gia có cùng thứ nguyên nhỏ hơn 𝑟 (giới hạn trên của xếp hạng). Độ phức tạp về thời gian và không gian đối với mạng chéo là 𝑂 (𝑑 2𝐿𝑐) và đối với hỗn hợp DCN cấp thấp (DCN-Mix) thì hiệu quả khi 𝑟𝐾 ≪ 𝑑 với 𝑂 (2𝑑𝑟𝐾𝐿).

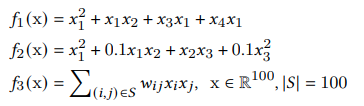
### 2.4.5 Kỹ thuật chéo đặc trưng

Nhiều công trình gần đây đã đề xuất mô hình hóa các điểm giao nhau rõ ràng không thể học được một cách hiệu quả từ các mạng nơ-ron truyền thống. Tuy nhiên, hầu hết các công trình chỉ nghiên cứu các tập dữ liệu công cộng với các mẫu chéo không xác định và dữ liệu nhiễu; rất ít công trình được nghiên cứu trong một bối cảnh sạch sẽ với các mô hình chân lý nền tảng đã biết. Do đó, điều quan trọng là phải hiểu: 1) trong những trường hợp nào thì mạng nơ ron truyền thống sẽ trở nên kém hiệu quả; 2) vai trò của từng thành phần trong mạng chéo của DCN-V2.

Chúng tôi sử dụng mạng chéo trong các mô hình DCN để đại diện cho các phương pháp chéo đặc trưng đó và so sánh với các ReLU, được sử dụng phổ biến trong các hệ thống khuyến nghị công nghiệp. Để đơn giản hóa các thí nghiệm và dễ hiểu, chúng tôi giả sử mỗi đối tượng 𝑥𝑖 có chiều một, và đơn thức 𝑥 𝛼1 1 𝑥 𝛼2 2 · · · 𝑥 𝛼𝑑 𝑑 đại diện cho một | 𝜶 | - thứ tự tương tác giữa các đối tượng.

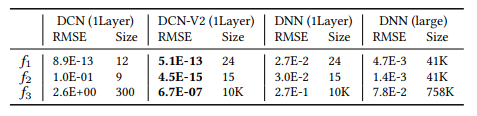
#### 2.4.5.1 Hiệu suất với độ khó tăng dần

Chỉ xem xét các điểm giao nhau của đối tượng bậc 2 và cho mô hình chân trị cơ bản là 𝑓 (x) = Í | 𝜶 | = 2 𝑤𝜶 𝑥 𝛼1 1 𝑥 𝛼2 2. . . 𝑥 𝛼𝑑 𝑑. Sau đó, độ khó của việc học 𝑓 (x) phụ thuộc vào: 1) độ thưa thớt (𝑤𝜶 = 0), số phép lai và 2) độ giống nhau của các mẫu chữ thập (được đặc trưng bởi Var (𝑤𝜶)), nghĩa là sự thay đổi trong một tính năng sẽ đồng thời ảnh hưởng đến hầu hết các điểm chéo tính năng với số lượng tương tự. Chúng tôi tạo bộ dữ liệu tổng hợp với độ khó ngày càng tăng trong Eq. (6).



trong đó tập hợp 𝑆 và trọng số 𝑤𝑖𝑗 được gán ngẫu nhiên, và các trọng số 𝑥𝑖 được lấy mẫu đồng nhất từ ​​khoảng [-1, 1].

Báo cáo Bảng 1 có nghĩa là RMSE trong số 5 lần chạy và kích thước mô hình. Khi các mẫu chữ thập là đơn giản (𝑓1), cả DCN-V2 và DCN đều hiệu quả. Khi các mẫu trở nên phức tạp hơn (𝑓3), DCNV2 vẫn chính xác trong khi DCN suy giảm. Hiệu suất của DNN vẫn kém ngay cả khi có cấu trúc rộng hơn và sâu hơn (kích thước lớp [200, 200] cho 𝑓1 và 𝑓2, [1024, 512, 256] cho 𝑓3). Điều này cho thấy sự kém hiệu quả của DNN trong việc mô hình hóa các mẫu đơn thức.



#### 2.4.5.2 Vai trò của từng thành phần

Chúng tôi cũng đã tiến hành các nghiên cứu cắt bỏ trên các đa thức thuần nhất bậc 3 và 4 tương ứng. Đối với mỗi đơn hàng, chúng tôi chọn ngẫu nhiên 20 số hạng chéo từ x ∈ R 50.

Hình 4 cho thấy sự thay đổi RMSE trung bình theo độ sâu lớp. Rõ ràng, x0 ⊙ (𝑊 x𝑖) mô hình trật tự-𝑑 giao nhau ở lớp 𝑑-1, được xác minh rằng hiệu suất tốt nhất cho đa thức bậc 3 đạt được ở lớp 2 (tương tự cho bậc 4). Tuy nhiên, ở các lớp khác, hiệu suất giảm đáng kể. Đây là nơi mà các điều khoản thiên vị và điều khoản còn lại rất hữu ích - chúng tạo và duy trì tất cả các dấu thập theo thứ tự cao nhất. Điều này làm giảm khoảng cách hiệu suất giữa các lớp và ổn định mô hình khi các dấu chéo dự phòng được đưa vào. Điều này đặc biệt quan trọng đối với các ứng dụng trong thế giới thực với các mẫu chéo không xác định.

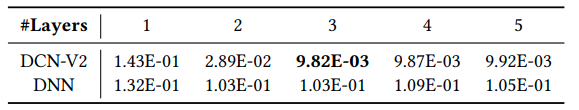
#### 2.4.5.3 Hiệu suất với độ sâu lớp ngày càng tăng

Bây giờ chúng tôi nghiên cứu các tình huống gần với cài đặt thế giới thực hơn, trong đó các thuật ngữ chéo có thứ tự kết hợp.



trong đó tập hợp được chọn ngẫu nhiên 𝑆 = 𝑆2 ∪ 𝑆3 ∪ 𝑆4, | 𝑆2 | = 20, | 𝑆3 | = 10, | 𝑆4 | = 5, và ∀𝜶 ∈ 𝑆𝑖, | 𝜶 | = 𝑖; sin giới thiệu nhiễu loạn và đại diện cho tiếng ồn Gaussian.

Bảng báo cáo RMSE trung bình trong số 5 lần chạy. Với sự gia tăng độ sâu của lớp, CN-M có thể thu thập các giao diện tính năng bậc cao hơn trong dữ liệu, dẫn đến cải thiện hiệu suất. Nhờ các điều khoản thiên vị và dư thừa, hiệu suất không bị suy giảm sau lớp 3, nơi các tương tác tính năng dư thừa được giới thiệu.



Tóm lại, ReLU không hiệu quả trong việc thu thập các đối tượng địa lý rõ ràng (quan hệ nhân) ngay cả với một mạng sâu hơn và lớn hơn. Điều này phù hợp tốt với các nghiên cứu trước đây [1]. Độ chính xác giảm đi đáng kể khi các mẫu chữ thập trở nên phức tạp hơn. DCN chụp chính xác các mẫu chữ thập đơn giản nhưng không thành công ở các mẫu phức tạp hơn. Mặt khác, DCN-V2 vẫn chính xác và hiệu quả đối với các mẫu chữ thập phức tạp.

### 2.4.6 Phân tích mô hình

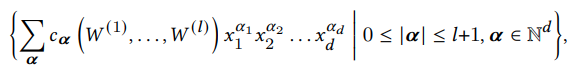
Phần này phân tích DCN-V2 từ quan điểm xấp xỉ đa thức và tạo kết nối với công việc liên quan.

Kí hiệu. Cho vectơ nhúng x = [x1; x2; . . . ; x𝑘] = [𝑥1, 𝑥2 ,. . . , 𝑥𝑑] ∈ R 𝑑 là một vectơ cột, trong đó x𝑖 ∈ R 𝑒𝑖 đại diện cho việc nhúng đối tượng thứ và 𝑥𝑖 đại diện cho phần tử thứ trong x. Cho đa chỉ số 𝜶 = [𝛼1, · · ·, 𝛼𝑑] ∈ N 𝑑 và | 𝜶 | = Í𝑑 𝑖 = 1 𝛼𝑖. 𝐶 𝑏 𝑎 B y ∈ {1, · · ·, 𝑎} 𝑏 ∀𝑖 <𝑗, 𝑦𝑖> 𝑦𝑗. Gọi 1 là vectơ của tất cả các 1 và 𝐼 là ma trận nhận dạng. Chúng tôi sử dụng chữ in hoa cho ma trận, chữ in thường đậm cho vectơ và chữ thường cho đại lượng vô hướng.

#### 2.4.6.1 Xấp xỉ đa thức

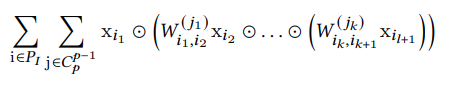
Chúng tôi phân tích DCN-V2 từ hai quan điểm của xấp xỉ đa thức - 1) Coi mỗi phần tử (bit) 𝑥𝑖 là một đơn vị, và phân tích tương tác giữa các phần tử (Định lý 4.1); và 2) Coi mỗi đối tượng nhúng x𝑖 là một đơn vị, và chỉ phân tích các tương tác khôn ngoan của đối tượng địa lý (Định lý 4.2) (chứng minh trong Phụ lục).

Định lý 4.1 (Bitwise). Giả sử đầu vào của một mạng chéo 𝑙 lớp là x ∈ R 𝑑, đầu ra là 𝑓𝑙 (x) = 1 ⊤x 𝑙 và tầng thứ được định nghĩa là x 𝑖 = x ⊙ 𝑊 (𝑖 − 1) x 𝑖 −1 + x 𝑖 − 1. Sau đó, đa thức nhiều biến 𝑓𝑙 (x) tái tạo các đa thức trong lớp sau:





Định lý 4.2 (đặc trưng). Với cài đặt tương tự như trong Định lý 4.1, chúng tôi giả sử thêm đầu vào x = [x1; . . . ; x𝑘] chứa 𝑘 tính năng nhúng và coi mỗi x𝑖 là một đơn vị. Sau đó, đầu ra x 𝑙 của một mạng chéo 𝑙 lớp tạo ra tất cả các tương tác đối tượng theo thứ tự 𝑙 + 1. Cụ thể, đối với các đối tượng có chỉ số (lặp lại) trong 𝐼, hãy đặt 𝑃𝐼 = 𝑃𝑒𝑟𝑚𝑢𝑡𝑎𝑡𝑖𝑜𝑛𝑠 (𝐼), sau đó thứ tự của chúng- 𝑝 tương tác được đưa ra bởi:

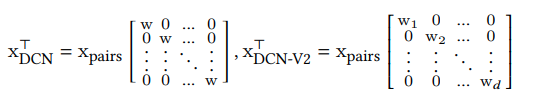


Từ cả góc độ bitwise và quan điểm về tính năng, mạng chéo có thể tạo tất cả các tương tác tính năng theo thứ tự 𝑙 + 1 cho một mạng chéo nhiều lớp. So với DCN-V, DCN-V2 đặc trưng cho cùng một lớp đa thức với nhiều tham số hơn và biểu cảm hơn. Hơn nữa, các tương tác tính năng trong DCN-V2 biểu cảm hơn và có thể được xem theo cả chiều dọc lẫn tính năng, trong khi trong DCN thì chỉ theo chiều dọc theo chiều bit.

#### 2.4.6.2 Kết nối với công việc liên quan

Chúng tôi nghiên cứu mối liên hệ giữa DCN-V2 và các phương pháp học tập tương tác tính năng khác của SOTA; chúng ta chỉ tập trung vào thành phần tương tác tính năng của từng mô hình và bỏ qua thành phần DNN. Đối với mục đích so sánh, chúng tôi giả định rằng các nhúng đối tượng địa lý có kích thước bằng nhau 𝑒.

DCN. Mô hình đề xuất của chúng tôi phần lớn được lấy cảm hứng từ DCN. Hãy xem chế độ xem hình chiếu hiệu quả của DCN, tức là, nó hoàn toàn tạo ra tất cả các điểm giao nhau theo cặp và sau đó chiếu nó sang một không gian có chiều thấp hơn; DCN-V2 tương tự với cấu trúc chiếu khác.



trong đó xpairs = [𝑥𝑖𝑥˜𝑗] ∀𝑖, 𝑗 chứa tất cả 𝑑 2 tương tác theo cặp giữa x0 và ˜x; w ∈ R 𝑑 là vectơ trọng số trong DCN-V; w𝑖 ∈ R 𝑑 là cột thứ của ma trận trọng số trong DCN-V2 (Phương trình (1)).

DLRM và DeepFM. Cả hai về cơ bản đều là FM bậc 2 mà không có thành phần DNN (bỏ qua những khác biệt nhỏ). Do đó, chúng tôi đơn giản hóa việc phân tích và so sánh với FM có công thức x ⊤𝜷 + Í 𝑖 <𝑗 𝑤𝑖𝑗⟨x𝑖, x𝑗⟩. Điều này tương đương với DCN-V2 1 lớp (Phương trình (1) không có số dư) với ma trận trọng lượng có cấu trúc.

