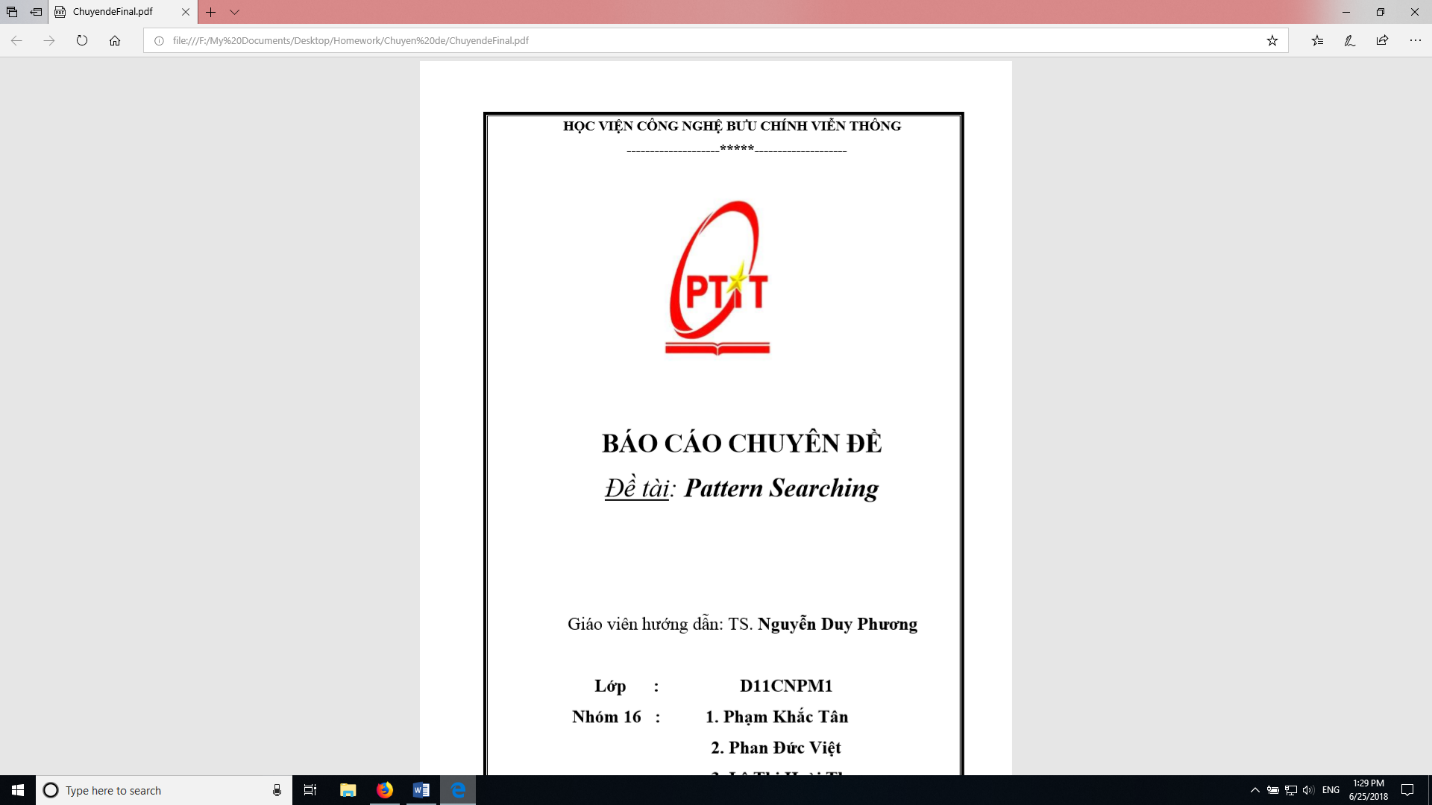
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

**-----\*\*\*\*\*-----**

**BÁO CÁO**

**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**KẾT HỢP HỌC SÂU VÀ LỌC CỘNG TÁC CHO HỆ TƯ VẤN SÁCH**

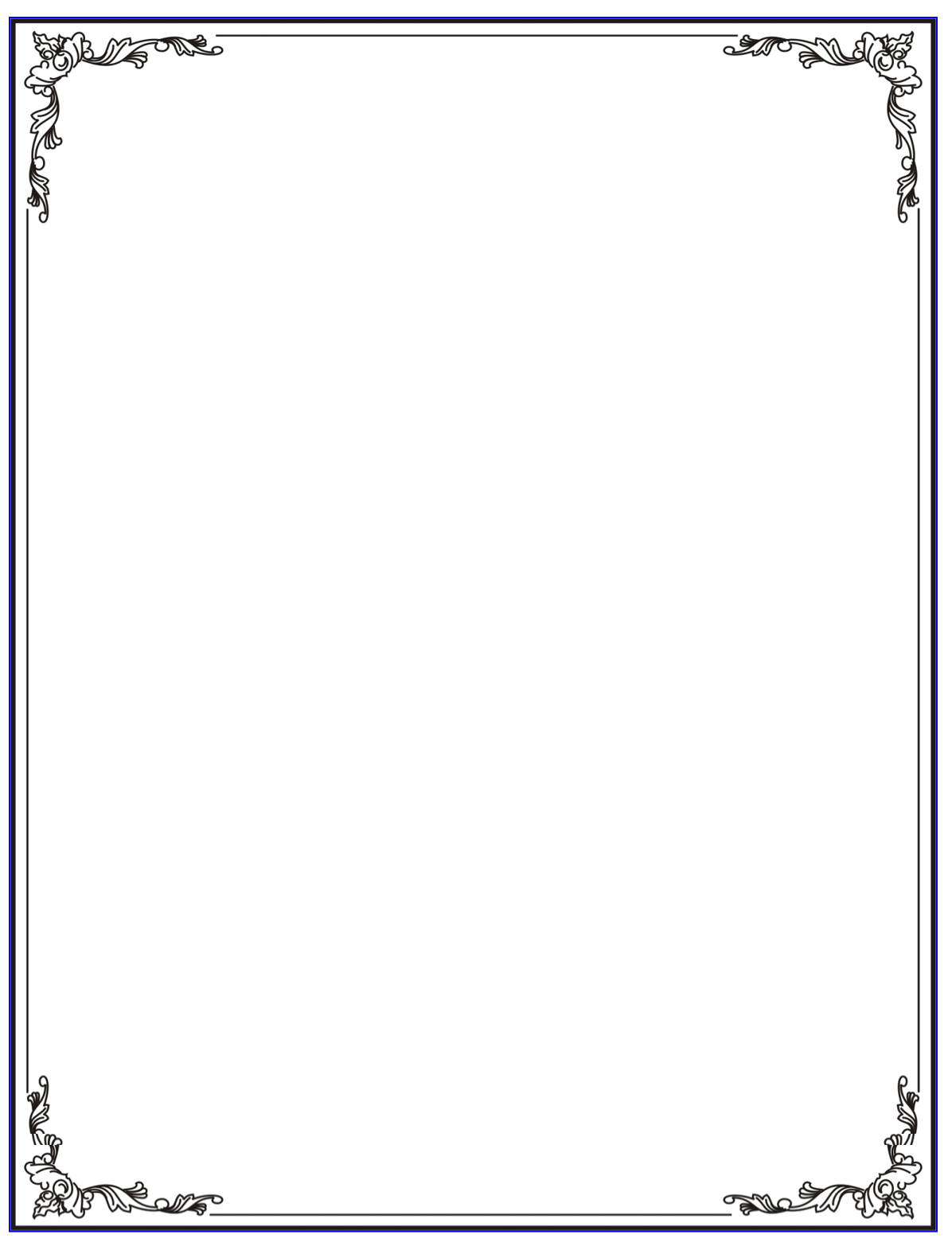
**Giảng viên: Trần Đình Quế**

**Sinh viên: Nguyễn Hữu Mừng**

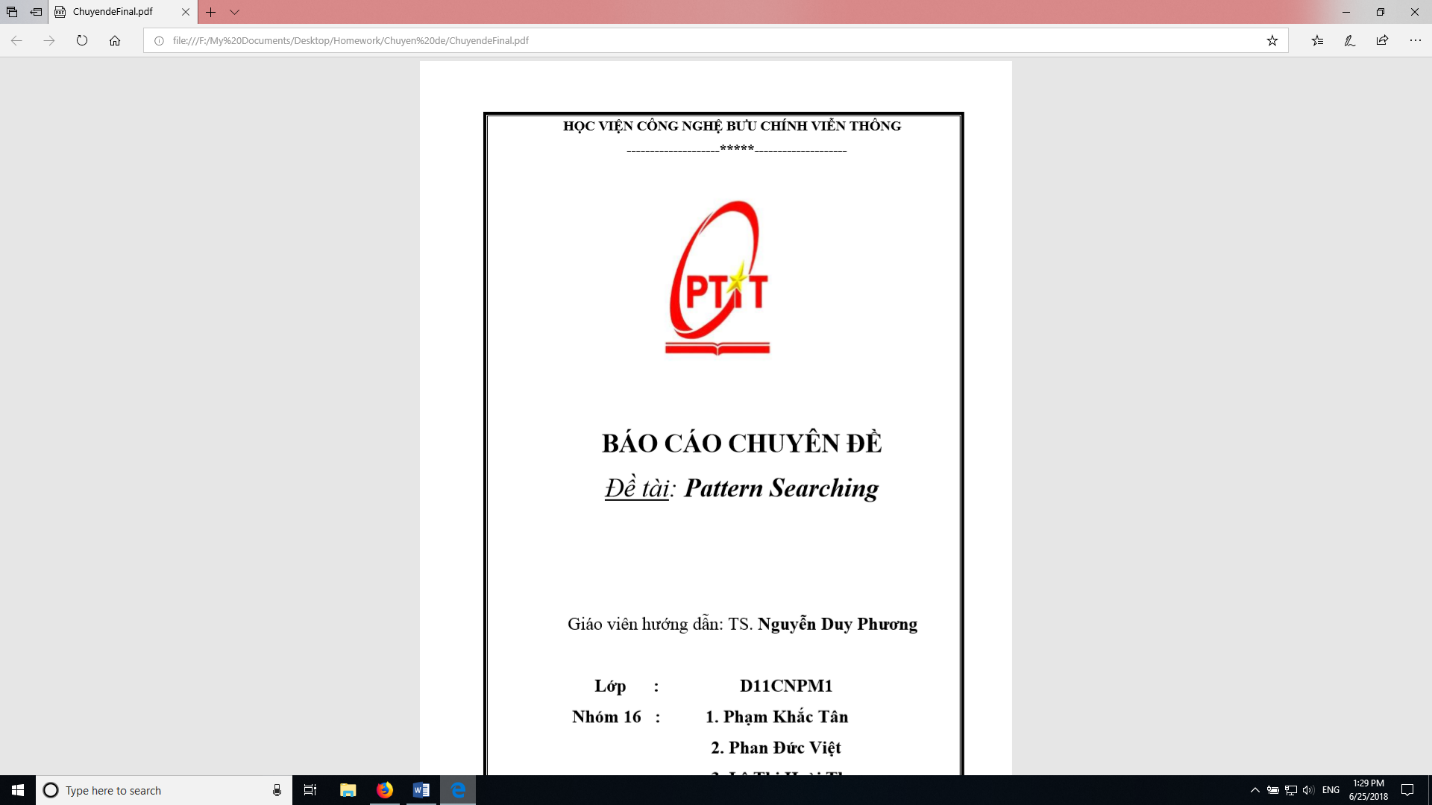
**Mã sinh viên: B18DCCN419**

**Lớp: D18CNPM1**

**Hà Nội - 2022**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

**-----\*\*\*\*\*-----**

**BÁO CÁO**

**ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**KẾT HỢP HỌC SÂU VÀ LỌC CỘNG TÁC CHO HỆ TƯ VẤN SÁCH**

**Giảng viên: Trần Đình Quế**

**Sinh viên: Nguyễn Hữu Mừng**

**Mã sinh viên: B18DCCN419**

**Lớp: D18CNPM1**

**Hà Nội - 2022**

## LỜI CẢM ƠN

Lời đầu tiên để có thể hoàn thành đồ án tốt nghiệp này em xin chân thành cảm ơn tới thầy Trần Đình Quế đã tận tình hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn tới các thầy cô trong Học viện và khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức rất quan trọng. Đó là nền tảng để em có thể áp dụng vào thực hiện đồ án cũng như công việc sau khi ra trường.

Lời cảm ơn cuối cùng em muốn gửi tới gia đình, bạn bè đã giúp đỡ, quan tâm em trong suốt quá trình học tập và làm đồ án tốt nghiệp.

Do kiến thức và kinh nghiệm còn hạn chế nên mặc dù đã cố gắng, đồ án của em vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của thầy/cô để đồ án hoàn thiện hơn.

Em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng 12 năm 2022* |
|  | **Sinh viên** |
|  |  |
|  | **Nguyễn Hữu Mừng** |

**Nhận xét, đánh giá, cho điểm**

**(Của giảng viên hướng dẫn)**

………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….Điểm: …………………………………………..(bằng chữ: …………………………………)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ đồ án trước hội đồng chấm tốt nghiệp

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 12 năm 2022 |
|  | **Giảng viên hướng dẫn** |
|  |  |
|  | **PGS.TS. Trần Đình Quế** |

**Nhận xét, đánh giá, cho điểm**

**(Của giảng viên phản biện)**

………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….………………………………………………………………………………………………….Điểm: …………………………………………..(bằng chữ: …………………………………)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ đồ án trước hội đồng chấm tốt nghiệp

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 12 năm 2022 |
|  | **Giảng viên phản biện** |
|  |  |
|  |  |

## MỤC LỤC

[LỜI CẢM ƠN i](#_Toc117433730)

[MỤC LỤC iv](#_Toc117433731)

[DANH MỤC TỪ VIẾT TẮT vi](#_Toc117433732)

[DANH MỤC CÁC HÌNH ẢNH vii](#_Toc117433733)

[DANH MỤC CÁC BẢNG viii](#_Toc117433734)

[MỞ ĐẦU 1](#_Toc117433735)

[CHƯƠNG 1 - Xác định bài toán 2](#_Toc117433736)

[CHƯƠNG 2 - Một số kỹ thuật … 3](#_Toc117433737)

[1. MF 3](#_Toc117433738)

[2. MLP: 3](#_Toc117433739)

[Embedding 3](#_Toc117433740)

[Dense 3](#_Toc117433741)

[Dropout: 3](#_Toc117433742)

[BatchNormalization: 3](#_Toc117433743)

[3. Kết hợp MLP với MF: 3](#_Toc117433744)

[4. Một số hàm kích hoạt: 4](#_Toc117433745)

[Hàm sigmoid: 4](#_Toc117433746)

[Hàm tanh: 5](#_Toc117433747)

[Hàm reLU: 6](#_Toc117433748)

[Hàm Leaky reLU: 7](#_Toc117433749)

[5. Loss function và một số thuật toán tối ưu: 8](#_Toc117433750)

[5.1. Loss function: 8](#_Toc117433751)

[5.2. Một số thuật toán tối ưu: 9](#_Toc117433752)

[5.2.2. Stochastic Gradient Descent (SGD): 10](#_Toc117433753)

[5.2.3. Adagrad: 10](#_Toc117433754)

[5.2.4. RMSprop: 11](#_Toc117433755)

[5.2.5. Adam: 11](#_Toc117433756)

[CHƯƠNG 3 - Thử nghiệm, đánh giá, tích hợp hệ thống 12](#_Toc117433757)

[1. Thử nghiệm, đánh giá: 12](#_Toc117433758)

[1.1. Tập dữ liệu: 12](#_Toc117433759)

[1.2. Chỉ số đánh giá mô hình: 13](#_Toc117433760)

[1.3. Thử nghiệm một số mô hình thuật toán: 15](#_Toc117433761)

[2. Một số giao diện của hệ thống: 16](#_Toc117433762)

[KẾT LUẬN 19](#_Toc117433763)

[1. Kết luận: 19](#_Toc117433764)

[2. Hướng phát triển: 19](#_Toc117433765)

[TÀI LIỆU THAM KHẢO 20](#_Toc117433766)

## DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Viết đầy đủ** | **Giải nghĩa** |
| NCF | Neural network-based Collaborative Filtering | Mạng nơ-ron dựa trên lọc cộng tác |
| MF | Matrix Factorization | Phân tích ma trận thành nhân tử |
| lr | Learning rate | Tốc độ học |
| GD | Gradient Descent |  |
| SGD | Stochastic Gradient Descent |  |
| ID | Identification |  |
| HR | Hit Ratio |  |
| NDCG | Normalized Discounted Cumulative Gain |  |
|  |  |  |
|  |  |  |

## DANH MỤC CÁC HÌNH ẢNH

[Hình 3.1: minh họa một số hàng dữ liệu trong file ratings.csv 7](#_Toc116846653)

[Hình 3.2: minh họa một số hàng dữ liệu cần thiết trong file books.csv 8](#_Toc116846654)

[Hình 3.3: Giao diện đăng nhập 11](#_Toc116846655)

[Hình 3.4: Giao diện đăng ký 11](#_Toc116846656)

[Hình 3.5: Giao diện quên mật khẩu 12](#_Toc116846657)

[Hình 3.6: Giao diện quên mật khẩu 12](#_Toc116846658)

## DANH MỤC CÁC BẢNG

[Bảng 3.1: kết quả thử nghiệm với optimizer khác nhau 4](#_Toc116827389)

# MỞ ĐẦU

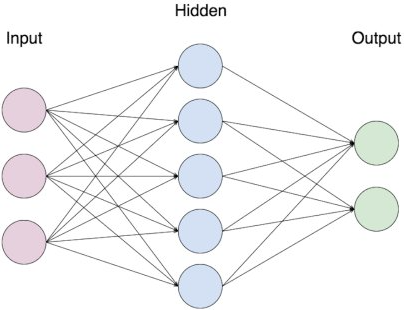
# Xác định bài toán

# Một số kỹ thuật …

## MF

## MLP:

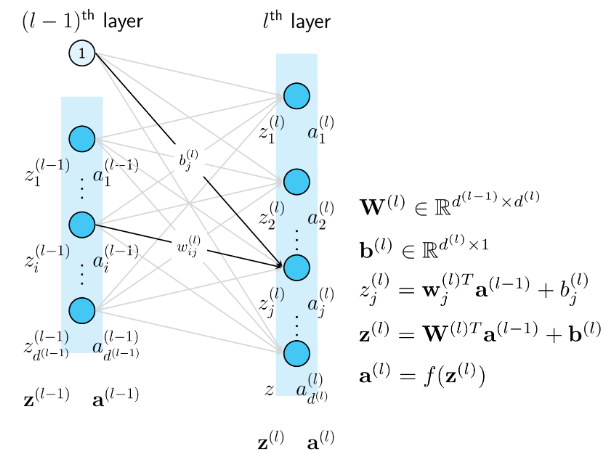
Là sự kết hợp của các tầng perceptron hay còn được gọi là perceptron đa tầng (multilayer perceptron - MLP)



Một mạng Neural sẽ có 3 kiểu tầng:

* Tầng vào (input layer): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
* Tầng ra (output layer): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
* Tầng ẩn (hidden layer): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.

Một node hình tròn trong một layer được gọi là một unit. Unit ở các input layer, hidden layers, và output layer được lần lượt gọi là input unit, hidden unit, và output unit. Đầu vào của các hidden layer được ký hiệu bởi z, đầu ra của mỗi unit thường được ký hiệu là a (thể hiện activation, tức giá trị của mỗi unit sau khi ta áp dụng activation function lên z). Đầu ra của unit thứ i trong layer thứ l được ký hiệu là  Giả sử thêm rằng số unit trong layer thứ l (không tính bias) là d(l) Vector biểu diễn output của layer thứ l được ký hiệu là a(l) ∈



Trong đó:

* **W**(l) là trọng số tại tầng thứ l
* **b**(l) là bias tại tầng thứ l.
* **z**(l) là đầu ra tại tầng thứ l
* **a**(l) là giá trị đầu ra của hàm kích hoạt sau khi áp dụng lên **z**

Mỗi tầng trong mạng neural có thể áp dụng bởi kiến trúc liên kết mạng khác nhau. Tiếp theo ta sẽ tìm hiểu một số kiểu layer trong MLP:

### Embedding layer:

Là một kỹ thuật đưa một vector có số chiều lớn, thường ở dạng thưa, về một vector có số chiều nhỏ, thường ở dạng dày đặc. Phương pháp này đặc biệt hữu ích với những đặc trưng hạng mục có số phần tử lớn ở đó phương pháp chủ yếu để biểu diễn mỗi giá trị thường là một vector dạng one-hot.

Một cách lý tưởng, các giá trị có ý nghĩa tương tự nhau nằm gần nhau trong không gian embedding. Không gian véc-tơ này phải thể hiện được bản chất của tập dữ liệu ban đầu và cực tiểu hóa lượng mất mát thông tin xảy ra khi bạn chuyển sang không gian mới.

<https://stats.stackexchange.com/questions/270546/how-does-keras-embedding-layer-work>

### **Dense layer (full connect layer)**:

Là một lớp cổ điển trong mạng [nơ ron](https://trituenhantao.io/tu-dien-thuat-ngu/no-ron/) nhân tạo. Mỗi nơ ron nhận đầu vào từ tất cả nơ ron lớp trước đó.

### Dropout layer:

Kỹ thuật dropout là việc chúng ta sẽ bỏ qua một vài unit trong suốt quá trình huấn luyện trong mô hình, những unit bị bỏ qua được lựa chọn ngẫu nhiên. Những unit này sẽ không tham gia và đóng góp vào quá trình lan truyền tiến và ngược.

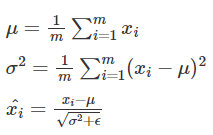
Mục đích của tầng Dropout để tránh over fitting và việc loại bỏ một số nút mạng sẽ buộc mô hình phải tìm ra những đặc trưng mới hơn, có thể hữu ích hơn khi kết hợp với nhiều neural khác.

### Batch Normalization layer:

Là một kĩ thuật để đào tạo mạng học sâu.

Thường ở những vị trí đầu tiên của mô hình để đạt hiệu quả cao nhất. Mục đích chính của batch normalization đó là chuẩn hóa dữ liệu đầu vào ở các layer theo batch về phân phối chuẩn để quá trình gradient descent hội tụ nhanh hơn, đồng thời giúp tránh được “bùng nổ đạo hàm” hoặc “đạo hàm triệt tiêu”. Từ đây training ổn định hơn.

Giả sử có một mini-batch với các giá trị của từng quan sát trong batch là [x1, x2, x3, x4, …]. Khi đó Batch Normalization sẽ được xác định thông qua phép chuẩn hóa trung bình và phương sai của các phần tử trong batch theo công thức sau:



[https://viblo.asia/p/ai-interview-12-cau-hoi-phong-van-deep-learning-sieu-hay-khong-the-bo-qua-LzD5djvEZjY#\_1-trinh-bay-y-nghia-cua-batch-normalization-0](https://viblo.asia/p/ai-interview-12-cau-hoi-phong-van-deep-learning-sieu-hay-khong-the-bo-qua-LzD5djvEZjY%23_1-trinh-bay-y-nghia-cua-batch-normalization-0)

## Kết hợp MLP với MF:

<https://viblo.asia/p/tim-hieu-phuong-phap-loc-cong-tac-voi-mang-no-ron-ncf-4P856o2RKY3>

## Một số hàm kích hoạt:

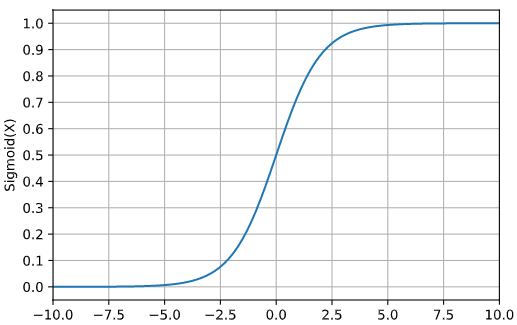
Trong [mạng neural nhân tạo](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_neural_nh%C3%A2n_t%E1%BA%A1o), hàm kích hoạt của một nút định nghĩa đầu ra của nút đó được cho bởi một đầu vào hay tập đầu vào. Đây đều là những hàm phi tuyến được áp dụng trên mỗi tầng ẩn của mạng neural. Nếu hàm kích hoạt là tuyến tính thì mạng nơ-ron của chúng ta dù có nhiều lớp vẫn sẽ có hiệu quả như một lớp tuyến tính mà thôi. Sự kết hợp của các activation functions giữa các tầng ẩn là để giúp mô hình học được các quan hệ phi tuyến phức tạp tiềm ẩn trong dữ liệu. Tùy vào từng bài toán, dữ liệu, đầu ra mong muốn và mỗi hàm kích hoạt đều có ưu nhược điểm khác nhau thì việc sử dụng nó sẽ khiến cho model ra kết quả khác nhau. Việc cần làm là phải chọn ra hàm kích hoạt phù hợp.

### Hàm sigmoid:

Công thức:



Đồ thị:



Đầu vào của hàm sigmoid là một số thực và đầu ra là giá trị nằm trong khoảng (0,1). Nhìn vào đồ thị: Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp. Tuy nhiên hiện nay hàm Sigmoid rất ít được dùng vì những nhược điểm sau:

*Hàm Sigmoid bão hòa và triệt tiêu gradient*:

Một nhược điểm dễ nhận thấy là khi đầu vào có trị tuyệt đối lớn (rất âm hoặc rất dương), gradient của hàm số này sẽ rất gần với 0 hoặc với 1. Khi đó hàm số đạt cực tiểu hoặc cực đại (hàm số bão hòa). Vì vậy hàm số bằng 0 tại điểm bão hòa, đạo hàm hàm số sẽ rất gần 0 dẫn đến việc cập nhật trọng số trong gradient descent trở nên vô nghĩa và hệ số neural network sẽ không học được nữa. Từ đây gây ra đạo hàm biến mất.

*Hàm Sigmoid không có trung tâm 0 gây khó khăn cho việc hội tụ:*

Do điểm bão hòa của hàm số là 0 và 1 nên có thể dễ dàng thấy được trung bình của hàm số không phải là 0. Output của hàm sigmoid luôn là số dương kết hợp với tính toán đạo hàm ở các tầng thì việc cập nhật trọng số sẽ luôn cùng dương hoặc cùng âm. Dẫn đến việc cập nhật sẽ chỉ diễn ra theo chiều hướng nhất định, hạn chế sự linh hoạt của mô hình.

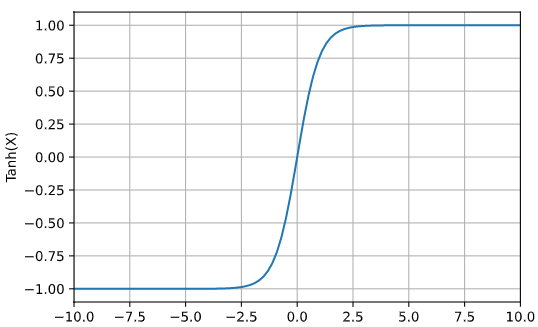
Để giải quyết vấn đề này ta có thể chuẩn hoá dữ liệu về dạng có trung tâm là 0 (zero-centered) với các thuật toán batch/layer normalization.

### Hàm tanh:

Công thức:



Đồ thị:



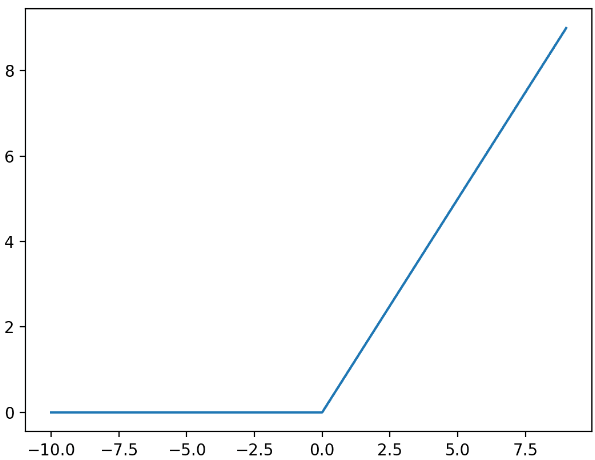
Đầu vào của hàm tanh là một số thực và đầu ra là giá trị nằm trong khoảng (0,1). Hàm này cũng có nhược điểm giống với hàm sigmoid ở chỗ nếu đầu vào có giá trị tuyệt đối lớn thì gradient sẽ rất nhỏ tiến dần tới giá trị bão hòa.

### Hàm reLU:

Công thức:



Đồ thị:



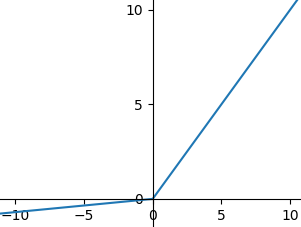
Gradient của hàm reLU được tính cực nhanh với gradient bằng 1 nếu đầu vào lớn hơn 0, bằng 0 nếu đầu vào nhỏ hơn 0. Đặc biệt hàm này không có đạo hàm tại điểm s = 0. Nhưng người ta vẫn thường định nghĩa reLU’(s) = 0 và khẳng định thêm rằng xác suất để input của một unit bằng 0 là rất nhỏ.

### Hàm Leaky reLU:

Công thức:



Đồ thị:

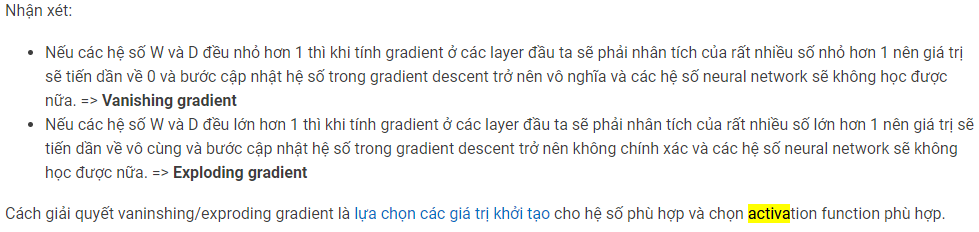


Hàm Leaky ReLU có các điểm tốt của hàm ReLU và giải quyết được vấn đề: Leaky ReLU tính đạo hàm cho các giá trị âm thay vì để giá trị là 0 đối với đầu vào nhỏ hơn 0.

<https://viblo.asia/p/mot-so-ham-kich-hoat-trong-cac-mo-hinh-deep-learning-tai-sao-chung-lai-quan-trong-den-vay-part-1-ham-sigmoid-bWrZn4Rv5xw>

<https://aicurious.io/posts/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks/>

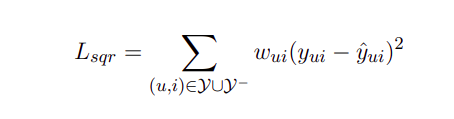
<https://nttuan8.com/bai-10-cac-ky-thuat-co-ban-trong-deep-learning/>



## 5. Loss function và một số thuật toán tối ưu:

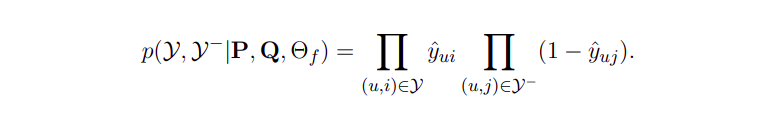
### 5.1. Loss function:

Thông thường, để tối ưu loss function ta dùng hàm độ lệch bình phương giữa điểm dự đoán và thực tế

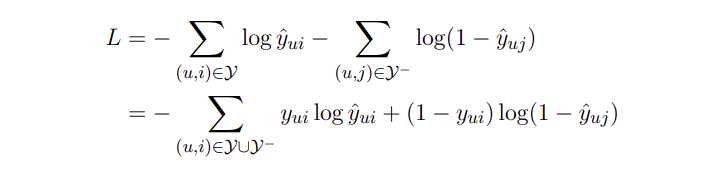


Đối với hàm này thường phù hợp cho các bài toán mà tập dữ liệu có phân phối Gauss, mà loại phân phối này thì thường không phù hợp với dữ liệu sinh ra từ tương tác người dùng và sản phẩm. Bởi vì tương tác người dùng chỉ có thể nhận một trong 2 giá trị 0 hoặc 1, 1 nếu có xảy ra tương tác, và 0 nếu không có bất cứ tương tác nào giữa người dùng và sản phẩm. Giá trị của hàm chỉ có thể là 0 hoặc 1, chính vì vậy, ta cần một độ đo khác đặc biệt và phù hợp cho loại dữ liệu như trên.

Đầu ra mô hình nằm trong đoạn [0, 1] nhờ hàm kích hoạt của lớp cuối cùng. Giá trị của đầu ra thể hiện độ liên quan giữa người dùng và sản phẩm. Điểm càng cao thì độ liên quan càng cao. Để học tham số của mô hình, ta sử dụng hàm likelihood có công thức như sau:



Lấy logarithm của hàm và đổi dấu, ta được:



Đây chính là hàm mục tiêu của các mô hình xây dựng theo phương pháp NCF. Hàm mục tiêu này có thể được tối ưu dựa vào các phương pháp tối ưu truyền thống như SGD hay Adam. Nó tương tự như hàm mục tiêu binary cross-entropy loss (hay còn gọi là log-loss).

### 5.2. Một số thuật toán tối ưu:

[https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8#\_1-gradient-descent-gd-2](https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8%23_1-gradient-descent-gd-2)

<https://machinelearningcoban.com/2017/01/16/gradientdescent2/>

<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>

<https://tek4.vn/khoa-hoc/machine-learning-co-ban/thuat-toan-toi-uu-adam>

### 5.2.2. Stochastic Gradient Descent (SGD):

Trong thuật toán này, tại 1 thời điểm, ta chỉ tính đạo hàm của loss function dựa trên chỉ một điểm dữ liệu rồi cập nhật trọng số dựa trên đạo hàm này. Với GD thông thường thì mỗi epoch ứng với 1 lần cập nhật θ, với SGD thì mỗi epoch ứng với N lần cập nhật θ với N là số điểm dữ liệu

Công thức cập nhật của SGD:



Trong đó J(θ;xi;yi) là loss function với chỉ 1 cặp điểm dữ liệu (input, label) là (xi;yi) và η là learning rate.

SGD phù hợp với bài toán có dữ liệu lớn và yêu cầu mô hình thay đổi liên tục (tức online learning). Ta hoàn toàn có thể kết hợp thêm thuật toán tăng tốc GD như momentum vào SGD.

### 5.2.3. Adagrad:

Không giống như thuật toán trước đó thì learning rate hầu như giống nhau trong quá trình training (learning rate là hằng số), Adagrad coi learning rate là 1 tham số. Tức là Adagrad sẽ cho learning rate biến thiên sau mỗi thời điểm t. Theo mặc định, learning rate của thuật toán này bằng 0.01. Công thức cập nhật của Adagrad:



Trong đó:

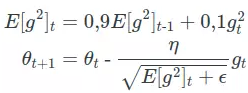
* Gt : gradient tại thời điểm t
* ϵ : hệ số tránh lỗi chia cho mẫu bằng 0
* G : là ma trận chéo mà mỗi phần tử trên đường chéo (i,i) là bình phương của đạo hàm vectơ tham số tại thời điểm t.

Một yếu điểm của thuật toán này là tổng bình phương biến thiên sẽ lớn dần theo thời gian cho đến khi nó đạt được tốc độ học rất nhỏ. Việc này làm cho traing có thể đóng băng.

### 5.2.4. RMSprop:

RMSprop giải quyết vấn đề tỷ lệ học giảm dần của Adagrad (vấn đề tốc độ học giảm dần theo thời gian sẽ khiến việc training chậm dần, có thể dẫn tới bị đóng băng) bằng cách chia tỷ lệ học cho trung bình của bình phương gradient.

Công thức ập nhật của RMSprop:



Thuật toán RMSprop có thể cho kết quả nghiệm chỉ là local minimum chứ không đạt được global minimum như Momentum. Vì vậy người ta sẽ kết hợp cả 2 thuật toán Momentum với RMSprop cho ra 1 thuật toán tối ưu Adam.

### 5.2.5. Adam:

Adam là sự kết hợp của Momentum và RMSprop. Đây là một thuật toán tối ưu hiệu quả trong học sâu.

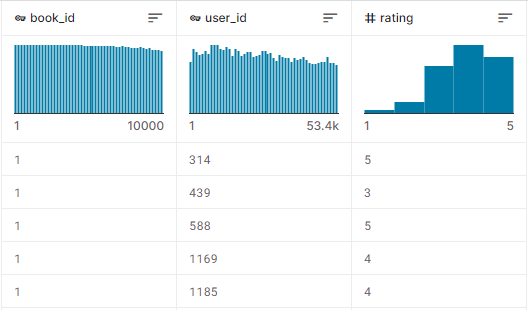
Tùy từng bài toán và dữ liệu thì việc áp dụng các thuật toán tối ưu khác nhau cho ra kết quả khác nhau. Đối với một bài toán thuật toán này có thể phù hợp cho kết quả tốt nhưng với bài toán khác, dữ liệu khác chưa chắc đã cho ra kết quả tốt như mong đợi. Ta sẽ thử nghiệm các thuật toán cho mô hình NCF cho hệ tư vấn sách ở chương sau.

# Thử nghiệm, đánh giá, tích hợp hệ thống

### 1. Thử nghiệm, đánh giá:

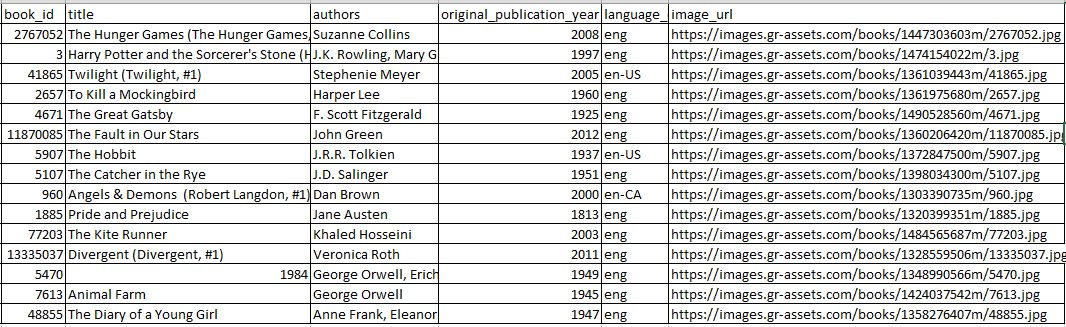
### 1.1. Tập dữ liệu:

* Tập dữ liệu được lấy từ trang *Kaggle.com* theo link sau: <https://www.kaggle.com/datasets/zygmunt/goodbooks-10k>
* Tập dữ liệu này chứa xếp hạng của mười nghìn cuốn sách phổ biến. Nói chung, có 100 đánh giá cho mỗi cuốn sách. Mỗi cuốn sách được đánh giá từ 1 đến 5
* Cả ID sách và ID người dùng đều là các số nguyên liên tiếp. ID sách được đánh số từ 1 đến 10000, ID người dùng đánh số từ 1 đến 53424. Tất cả người dùng đều thực hiện ít nhất 2 lượt đánh giá cho sách. Số lượng xếp hạng trung bình của mỗi người dùng là 8 lượt.
* Tập dữ liệu gồm có 6 file. Trong giới hạn đồ án này chỉ dùng đến 2 file là ratings.csv và books.csv.
* File ratings.csv gồm có 3 cột: user\_id, item\_id và rating:



Hình 3.1: minh họa một số hàng dữ liệu trong file ratings.csv

* File books.csv chứa 23 cột. Đối với hệ thống tư vấn sách trong đồ án này, chúng ta chỉ cần dùng đến các cột cần thiết: book\_id, authors, title, image\_url để hiển thị thông tin sách cho người dùng lên trang web.



Hình 3.2: minh họa một số hàng dữ liệu cần thiết trong file books.csv

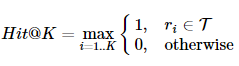
* Đầu vào của mô hình là vector user và item dạng one-hot. Chính vì vậy ta cần phải xử lí file ratings.csv…
* Tập train:
* Tập test:
* Đầu ra sẽ là item\_id thể hiện các sách nào được đề xuất cho người dùng, với mỗi id\_iem được dự đoán với giá trị nằm trong đoạn [0,1]. Giá trị càng gần 1 thì sách đó càng được đề xuất cao hơn cho người dùng, càng gần 0 cho biết sách đó nên được đề xuất ít cho người dùng.

### 1.2. Chỉ số đánh giá mô hình:

<https://cran.r-project.org/web/packages/recometrics/vignettes/Evaluating_recommender_systems.html>

* **Top K Hit Ratio (HR@K):**

Hit@K cho biết liệu có mục nào trong số K mục được đề xuất hàng đầu bởi mô hình NCF, có trong tập kiểm tra đối với một người dùng nhất định đã tương tác với mục đó hay không.



Trong đó, ri là mục được mô hình xếp hạng ở vị trí i trong số các mục tư vấn (các mục được sắp xếp theo thứ tự ưu tiên tư vấn giảm dần). T là tập hợp các mục mà người dùng đã tương tác nằm trong tập kiểm tra.

Tính trung bình giá trị Hit@K ta được chỉ số HR@K.

* **NDCG@K:**

Trong khi chỉ số trước đó chỉ xem xét trong K mục mà mô hình đề xuất liệu có hay không của một mục trong tập hợp thử nghiệm. Việc đánh giá này gây ra sự không công bằng cho các mục vì các mục được đề xuất ở vị trí ưu tiên khác nhau. NDCG nhằm mục đích đánh giá thứ hạng những giá trị này trong K mục được tư vấn đầu tiên. Trước hết tính chỉ số DCG@K:

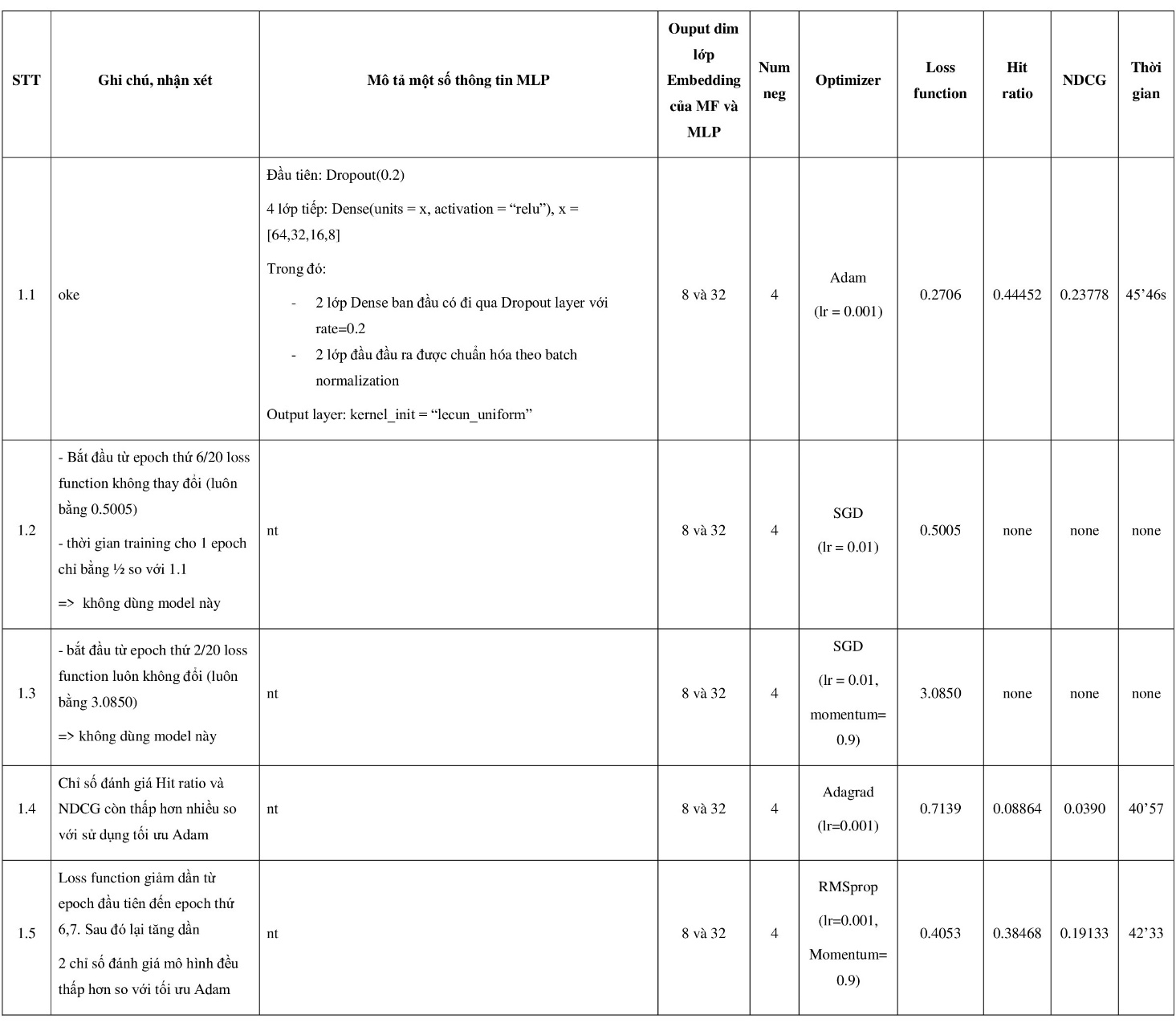


Trong đó Cri là giá trị tương tác quan sát được trong dữ liệu kiểm tra của mục ri và bằng 0 nếu các mục được đề xuất không chứa mục trong dữ liệu kiểm tra.

Khi đó NDCG@K được tính theo công thức:



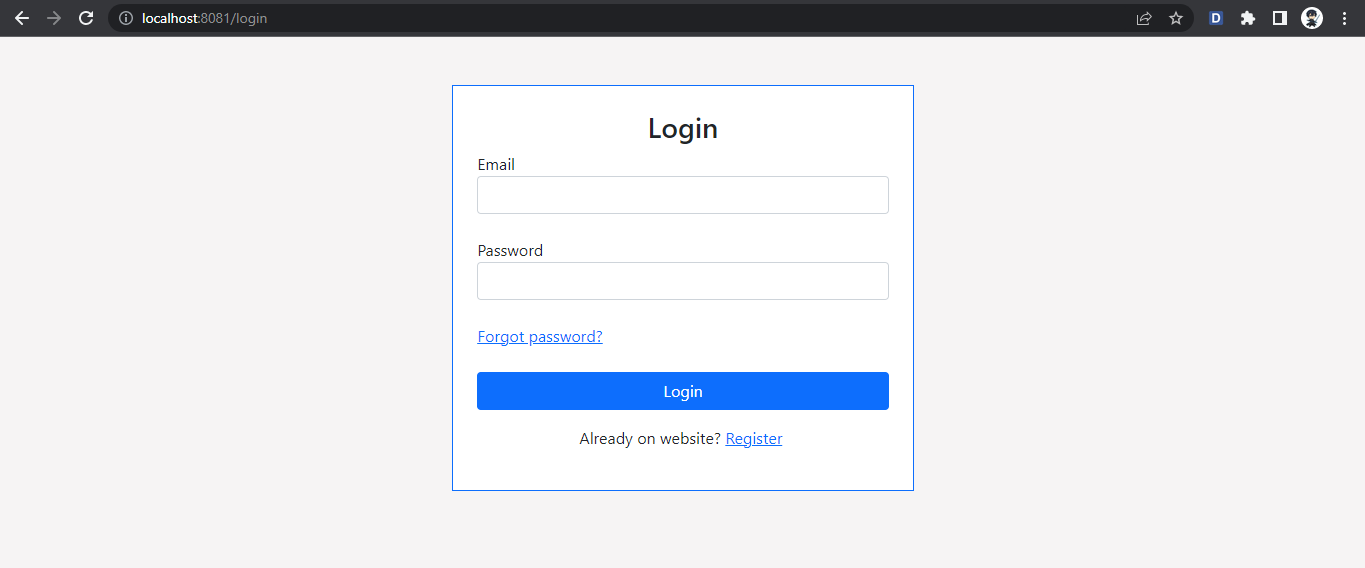
### 1.3. Thử nghiệm một số mô hình thuật toán:



Bảng 3.1: kết quả thử nghiệm với optimizer khác nhau

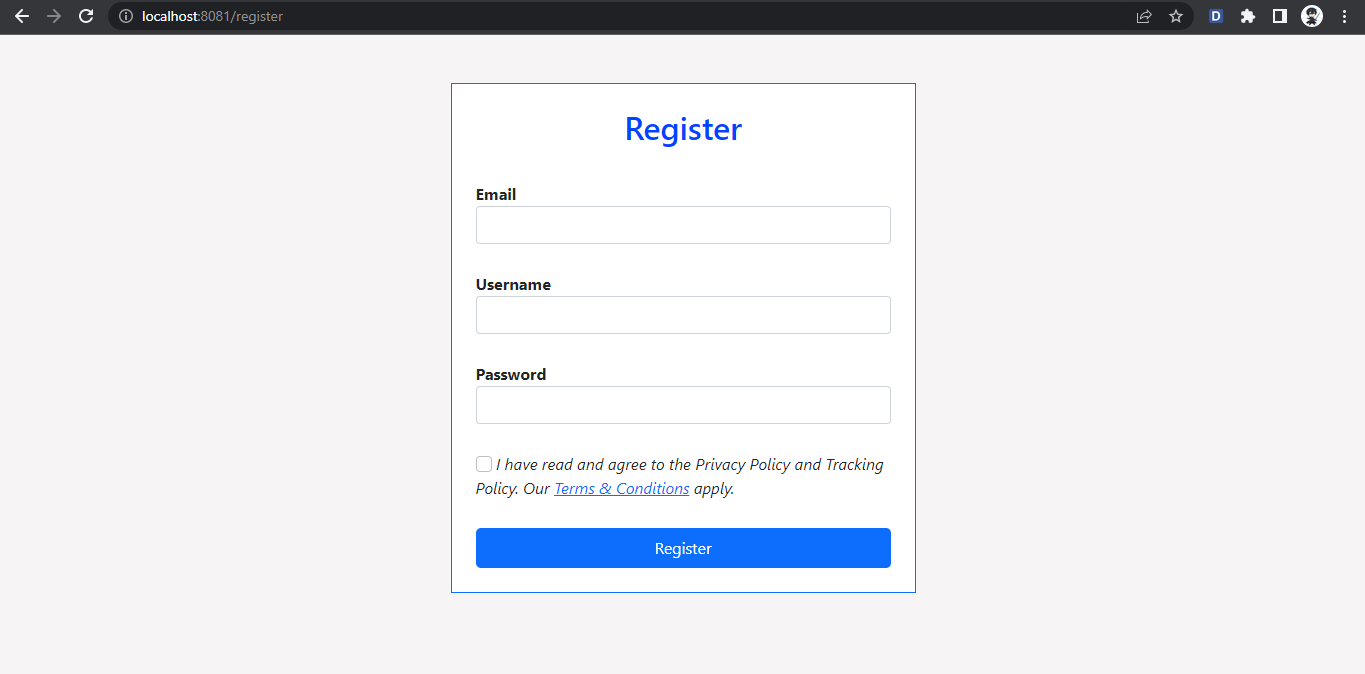
* Với bài toán này, dữ liệu như trên thì tối ưu Adam cho kết quả tốt nhất…..

### Một số giao diện của hệ thống:

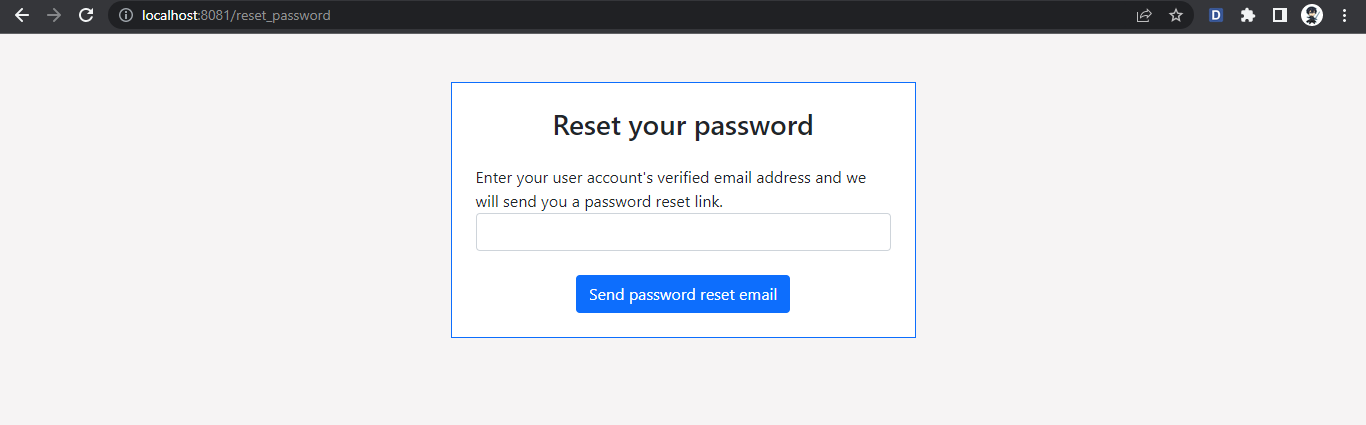


Hình 3.3: Giao diện đăng nhập

Giao diện cho phép người dùng nhập thông tin email, password để đăng nhập vào hệ thống. Tại đây người dùng cũng có thể chọn đăng ký hoặc chọn quên mật khẩu để tiến hành reset lại mật khẩu

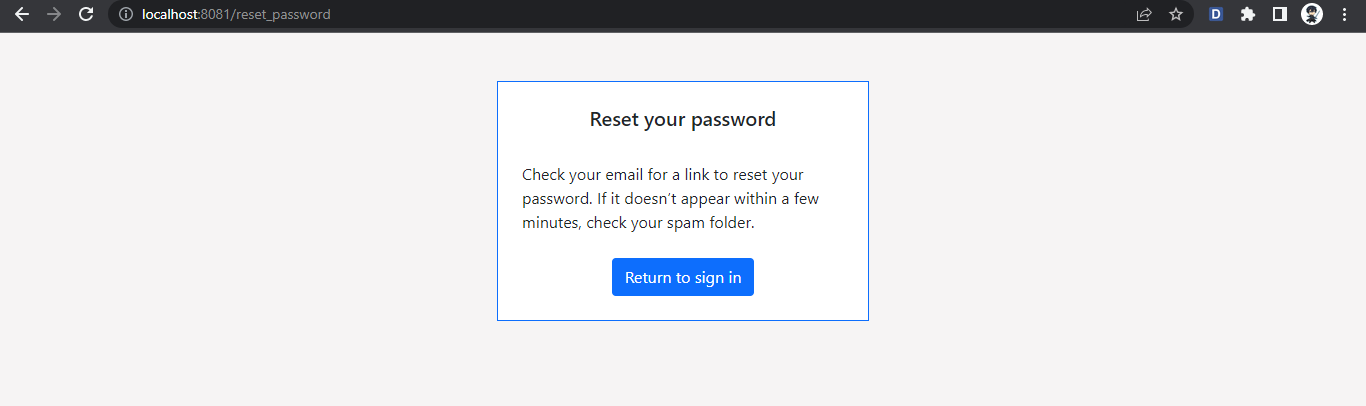


Hình 3.4: Giao diện đăng ký



Hình 3.5: Giao diện quên mật khẩu

Tại giao diện quên mật khẩu cho phép người dùng nhập email đã đăng ký trên hệ thống sau đó hệ thống sẽ gửi mail tới người dùng để reset lại mật khẩu. Sau khi người dùng bấm xác nhận gửi mail thì hiện giao diện như hình bên dưới.



Hình 3.6: Giao diện quên mật khẩu

# KẾT LUẬN

### Kết luận:

### Hướng phát triển:

* Về chức năng của ứng dụng:
* Về thuật toán tư vấn sách:

# TÀI LIỆU THAM KHẢO

1. M[artin T Hagan](https://www.amazon.com/Martin-T-Hagan/e/B001KE7YAO/ref=dp_byline_cont_book_1), [Howard B Demuth](https://www.amazon.com/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=Howard+B+Demuth&text=Howard+B+Demuth&sort=relevancerank&search-alias=books), [Mark H Beale](https://www.amazon.com/s/ref=dp_byline_sr_book_3?ie=UTF8&field-author=Mark+H+Beale&text=Mark+H+Beale&sort=relevancerank&search-alias=books), [Orlando De Jesús](https://www.amazon.com/s/ref=dp_byline_sr_book_4?ie=UTF8&field-author=Orlando+De+Jes%C3%BAs&text=Orlando+De+Jes%C3%BAs&sort=relevancerank&search-alias=books) (2014). Neural Network Design 2nd. Martin Hagan (September 1, 2014)
2. Charu C. Aggarwal (2016). Recommender Systems 1st. Springer (March 28, 2016)
3. Charu C. Aggarwal (2018). Neural Networks and Deep Learning. Springer International Publishing AG. Gewerbestrasse 11, 6330 Cham, Switzerland
4. JOHN D. KELLEHER (2019). Deep learning. The Massachusetts Institute of Technology (2019)
5. Michael Nielse (2015). Neural Networks and Deep Learning. Determination Press, 2015.
6. Jojo Moolayil (2018). Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python. Apress (2018)
7. Umberto Michelucci (2018). Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks. Apress (2018).

[https://phamdinhkhanh.github.io/2019/12/26/Sorfmax\_Recommendation\_Neural\_Network.html#1-%C6%B0u-%C4%91i%E1%BB%83m-c%E1%BB%A7a-recommendation-neural-network](https://phamdinhkhanh.github.io/2019/12/26/Sorfmax_Recommendation_Neural_Network.html%231-%C6%B0u-%C4%91i%E1%BB%83m-c%E1%BB%A7a-recommendation-neural-network)