

**TỔNG CÔNG TY VIỄN THÔNG VIETTEL**

**KHỐI CÔNG NGHỆ THÔNG TIN**

**PYC THỬ NGHIỆM ĐÁNH GIÁ CÁC MÔ HÌNH MACHINE LEARNING VÀ DEEP LEARNING**

**TÀI LIỆU THIẾT KẾ CHI TIẾT**

**Mã hiệu dự án:** **MyViettel\_0008**

**Mã hiệu tài liệu: MyViettel\_0008**

**<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** |
| --- | --- | --- | --- | --- | --- | --- |
| 20/4/2023 | Đánh giá các nhóm mô hình Machine learning và Deep learning | A | Hoàn thành | PTC, PKH | * Mô tả chi tiết, đánh giá performance mô hình |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**MỤC LỤC**

[**I. Mô tả bộ dữ liệu 7**](#_heading=h.ay2vm15aknsu)

[1.1. Bộ dữ liệu tự sinh nhóm Machine Learning 7](#_heading=h.oyo1y9mnb0qm)

[**II. Build nhóm mô hình Machine Learning 7**](#_heading=h.yzv2jznlgj9y)

[2.1 Cài đặt môi trường 7](#_heading=h.tzfbyefumwd7)

[2.1.1 Cài đặt anaconda3 7](#_heading=h.gl9s318iyyf9)

[2.1.2 Tạo môi trường ảo 7](#_heading=h.ag0v909xcbpk)

[2.1.3 Setup các thư viện 8](#_heading=h.rwfx9eulbnk8)

[2.2. Sinh tập dữ liệu 9](#_heading=h.acbj27l4s3cz)

[2.3. Xử lý tập dữ liệu 10](#_heading=h.5lnpv1bsbmm3)

[2.4. Huấn luyện các mô hình 10](#_heading=h.mzku2i7xtc9e)

[2.4.1 Quá trình huấn luyện mô hình Linear Regression 10](#_heading=h.ki9s19d0o619)

[2.4.2 Quá trình huấn luyện mô hình Logistic Regression 11](#_heading=h.o98f1b6oja5x)

[2.4.3 Quá trình huấn luyện mô hình SVM 11](#_heading=h.gjp3m2oz2rp)

[**III. Build nhóm mô hình Deep learning 11**](#_heading=h.lyny3gviq1n7)

[3.1 Build mô hình RecVAE 11](#_heading=h.f7v67ac5xkkl)

[3.1.1. Cài đặt môi trường 11](#_heading=h.8mclfmoyrkse)

[Tạo môi trường ảo 11](#_heading=h.bhllqp1qupu4)

[Setup các thư viện 12](#_heading=h.nyz74o8fif16)

[3.1.2. Sinh dữ liệu 13](#_heading=h.4dhkvg4aunuz)

[Do mô hình sử dụng input trực tiếp vào model sau khi qua xử lý dữ liệu là sparse matrix, số feature của input tỷ lệ thuận với max(user), max(item) nên chúng ta có thể sử dụng tập movielens với nhiều tập dữ liệu có số user, item lớn khác nhau như: 13](#_heading=h.gqlvep5fcgy4)

[● MovieLens 1M Dataset: 1 million sample, from 6000 users on 4000 movies. 13](#_heading=h.k2l0btl89zxb)

[● MovieLens 10M Dataset: 10 million sample, 100,000 tag, 10,000 movies , 72,000 users 13](#_heading=h.l0631yesjaj2)

[● MovieLens 20M Dataset: 20 million ratings and 465,000 tag, 27,000 movies, 138,000 users 13](#_heading=h.914s6jmalma8)

[3.1.3. Xử lý tập dữ liệu 13](#_heading=h.z9mbn7t20rrv)

[3.1.4. Quá trình huấn luyện mô hình 17](#_heading=h.a1iqoj22283k)

[Kết quả 18](#_heading=h.gh0ri9dzst9y)

[3.2 Build mô hình BERT4REC 23](#_heading=h.bl5r19m0ew47)

[3.2.1. Cài đặt môi trường 23](#_heading=h.cg1b38fxu48w)

[3.2.1.1. Tạo môi trường ảo 23](#_heading=h.j2yfahhi38na)

[3.2.1.2. Setup các thư viện 23](#_heading=h.qbutmrhyf2tw)

[Cài đặt grpcio 27](#_heading=h.kotje5l34225)

[Cài đặt các thư viện khác 28](#_heading=h.q7ebteehlsmx)

[3.2.2. Sinh tập dữ liệu 29](#_heading=h.c07dx7gdsuty)

[Do mô hình sử dụng input trực tiếp vào model sau khi qua xử lý dữ liệu là sparse matrix, số feature của input tỷ lệ thuận với max(user), max(item) nên chúng ta có thể sử dụng tập movielens với nhiều tập dữ liệu có số user, item lớn khác nhau như: 29](#_heading=h.nlhms3w7io1r)

[● MovieLens 100K Dataset : 100,000 sample, 1000 users, 1700 movies) 29](#_heading=h.diw66x82sm06)

[● MovieLens 1M Dataset: 1 million sample, from 6000 users on 4000 movies 29](#_heading=h.m5n8ouz0aj64)

[● MovieLens 10M Dataset: 10 million sample,100,000 tag applications, 10,000 movies , 72,000 users 29](#_heading=h.p52d2ktqybm2)

[● MovieLens 20M Dataset: 20 million ratings and 465,000 tag applications, 27,000 movies, 138,000 users 29](#_heading=h.e01jlpea52wx)

[3.2.3. Xử lý tập dữ liệu 29](#_heading=h.re7n7ratsvil)

[3.2.4. Quá trình huấn luyện mô hình 32](#_heading=h.a6awxbgdaqdo)

[3.3 Build mô hình CONTRASTIVE LEARNING 36](#_heading=h.3k03bez469j5)

[3.3.1. Cài đặt môi trường 36](#_heading=h.fujmlka140ts)

[3.3.1.1. Tạo môi trường ảo 36](#_heading=h.x67owq79rhk4)

[3.3.1.2. Setup các thư viện 37](#_heading=h.hnorurdbsoo2)

[3.3.2. Sinh tập dữ liệu 40](#_heading=h.daiow0ib4a2v)

[3.3.3. Xử lý tập dữ liệu 41](#_heading=h.8u0gnwiev2wu)

[3.3.4. Quá trình huấn luyện mô hình 43](#_heading=h.xcih00w7o4nz)

[3.4 Build mô hình LSTM 47](#_heading=h.2pou3t2ajaw)

[3.4.1. Sinh tập dữ liệu 47](#_heading=h.eojh9c8foa15)

[3.4.2. Xử lý tập dữ liệu 48](#_heading=h.pgu5ipw8yopa)

[3.4.3. Quá trình huấn luyện mô hình 50](#_heading=h.wo08hyr3qjgr)

[**\*\*IV. Fine-Tune và đánh giá performance nhóm mô hình Machine learning 51**](#_heading=h.kk505xybv1ke)

[4.1. Fine-Tune và đánh giá performance nhóm mô hình Linear Regression 51](#_heading=h.ihiw3z3spim8)

[4.1.1. Test mô hình 51](#_heading=h.ryqbzywuzagh)

[4.1.2.Đánh giá performance 51](#_heading=h.f2uuvcp733ac)

[4.3.2.1 Đánh giá performance lần 1 52](#_heading=h.wbqonlrbrpoy)

[4.2. Fine-Tune và đánh giá performance nhóm mô hình Logistic Regression 53](#_heading=h.qoje2qn8tg7c)

[4.2.1. Test mô hình 53](#_heading=h.ctm3yt542v6c)

[4.2.2.Đánh giá performance 54](#_heading=h.otin9nr9e416)

[4.3.2.1 Đánh giá performance lần 1 54](#_heading=h.ykz19coup2v7)

[4.3. Fine-Tune và đánh giá performance nhóm mô hình SVM 54](#_heading=h.j77j45grjk66)

[4.3.1. Test mô hình 54](#_heading=h.t0sdwdukbo7i)

[4.3.2.Đánh giá performance 55](#_heading=h.8cl40ftehppg)

[4.3.2.1 Đánh giá performance lần 1 55](#_heading=h.otbbkmigat8u)

[4.3.2.2 Đánh giá performance lần 2 56](#_heading=h.z65vfdho9uu0)

[**V. Fine-Tune và đánh giá performance nhóm mô hình Deep learning 58**](#_heading=h.gtafci2aqy11)

[5.1 Fine-Tune và đánh giá performance nhóm mô hình RecVAE 58](#_heading=h.grbqg2iwexki)

[5.1.1 Test Mô hình 58](#_heading=h.ccwxhujaqkr9)

[5.1.2. Đánh giá performance 59](#_heading=h.z9h7jkz21ihz)

[5.2. Fine-Tune và đánh giá performance nhóm mô hình BERT4REC 60](#_heading=h.23481q64bzpk)

[5.2.1 Test Mô hình 60](#_heading=h.wva72061igk4)

[5.2.2. Đánh giá performance 61](#_heading=h.2p2e881e3myi)

[5.2.2.1 Đánh giá performance lần 1 61](#_heading=h.dzsyodhpy91k)

[5.2.2.2. Đánh giá performance lần 2 63](#_heading=h.koqfc7w1g3my)

[5.2.2.3. Đánh giá performance lần 3 64](#_heading=h.xtljcq61463i)

[5.2.2.4. Đánh giá performance lần 4 66](#_heading=h.qaxddjgemekc)

[5.2.2.5. Đánh giá performance lần 5 67](#_heading=h.57tx6sqvk2ub)

[5.3. Fine-Tune và đánh giá performance nhóm mô hình Contrastive Learning 67](#_heading=h.2ej04vlx7193)

[5.3.1 Test Mô hình 67](#_heading=h.co6y6gcy694e)

[5.3.2. Đánh giá performance 68](#_heading=h.voqmritzvzo9)

[5.3.2.1 Đánh giá performance lần 1 69](#_heading=h.4bafusqibrc3)

[5.3.2.2 Đánh giá performance lần 2 70](#_heading=h.w8siwbq2vptq)

[5.4. Fine-Tune và đánh giá performance nhóm mô hình LSTM 72](#_heading=h.r4agpv4sr86p)

[5.4.1 Test Mô hình 72](#_heading=h.nrlonqcy2m95)

[5.4.2. Đánh giá performance 73](#_heading=h.5o9uv6ptkrw1)

[5.4.2.1. Đánh giá performance lần 1 73](#_heading=h.1f8hyc1fgnst)

[5.4.2.2. Đánh giá performance lần 2 75](#_heading=h.jsd2x2py4199)

[5.4.2.3. Đánh giá performance lần 3 76](#_heading=h.vg7mi26o4msn)

[5.4.2.4. Đánh giá performance lần 4 78](#_heading=h.36jh6h9gdt6l)

**TRANG KÝ**

Người lập: <Ngày>

<Chức danh>

Người xem xét: <Ngày>

<Chức danh>

Người xem xét: <Ngày>

<Chức danh>

Người phê duyệt: <Ngày>

<Chức danh>

# 

# I. Mô tả bộ dữ liệu

## 1.1. Bộ dữ liệu tự sinh nhóm Machine Learning

# II. Build nhóm mô hình Machine Learning

## 2.1 Cài đặt môi trường

### 2.1.1 **Cài đặt anaconda3**

Cài đặt bản anaconda phù hợp với hệ thống từ <https://repo.anaconda.com/archive/> :

VD: [Anaconda3-2023.03-1-Linux-x86\_64.sh](https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-x86_64.sh)

| $ bash Anaconda3-2023.03-1-Linux-x86\_64.sh  $ source ~/.bashrc |
| --- |

### 2.1.2 Tạo môi trường ảo

Vì cài đặt offline nên ta sử dụng lệnh "conda create -n yourenvname --clone root" để tạo môi trường mới với phiên bản python và các gói thư viện mặc định theo môi trường base được khởi tạo khi cài đặt hoàn tất anaconda3.

| $ conda create -n gem\_ml --clone root  Source: /root/anaconda3  Destination: /root/anaconda3/envs/linear\_reg  The following packages cannot be cloned out of the root environment:  - defaults/linux-64::conda-23.3.1-py310h06a4308\_0  - defaults/linux-64::conda-build-3.24.0-py310h06a4308\_0  - defaults/noarch::conda-token-0.4.0-pyhd3eb1b0\_0  - defaults/linux-64::navigator-updater-0.3.0-py310h06a4308\_0  - defaults/linux-64::anaconda-navigator-2.4.0-py310h06a4308\_0  Packages: 139  Files: 91  Downloading and Extracting Packages  Downloading and Extracting Packages  Preparing transaction: done  Verifying transaction: done  Executing transaction: done  #  # To activate this environment, use  #  # $ conda activate gem\_ml  #  # To deactivate an active environment, use  #  # $ conda deactivate |
| --- |

Kích hoạt môi trường vừa mới khởi tạo:

| $ conda activate gem\_ml |
| --- |

### 2.1.3 Setup các thư viện

Để cài đặt các thư viện offline, tải các file .whl cần thiết trên <https://pypi.org/>:

[scikit\_learn-1.2.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl](https://files.pythonhosted.org/packages/fa/1e/36d7609e84b50d4a2e5bc43cd5013d9ea885799e5813a1e9cf5bb1afd3f4/scikit_learn-1.2.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl) (9.6 MB [view hashes](https://pypi.org/project/scikit-learn/#copy-hash-modal-c09f2261-5107-44dc-b0e1-3ce82cacf3de))

[tzdata-2023.3-py2.py3-none-any.whl](https://files.pythonhosted.org/packages/d5/fb/a79efcab32b8a1f1ddca7f35109a50e4a80d42ac1c9187ab46522b2407d7/tzdata-2023.3-py2.py3-none-any.whl) (341.8 kB [view hashes](https://pypi.org/project/tzdata/#copy-hash-modal-34fbe7bc-132f-46dd-af37-eb410b6d6ad4))

[pandas-2.0.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl](https://files.pythonhosted.org/packages/a3/40/eca46f6af07a83ea3b8706586b2d8a28c01bdccee789d24f2ccc5e148b28/pandas-2.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl) (12.3 MB [view hashes](https://pypi.org/project/pandas/#copy-hash-modal-4f1ebc2e-10f2-4521-a7c0-307aec7f1f76))

| $ pip install tzdata-2023.3-py2.py3-none-any.whl scikit\_learn-1.2.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl pandas-2.0.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl  Processing ./tzdata-2023.3-py2.py3-none-any.whl  Processing ./scikit\_learn-1.2.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl  Processing ./pandas-2.0.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl  Collecting joblib>=1.1.1  Using cached joblib-1.2.0-py3-none-any.whl (297 kB)  Collecting numpy>=1.17.3  Using cached numpy-1.24.3-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (17.3 MB)  Collecting scipy>=1.3.2  Using cached scipy-1.10.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (34.4 MB)  Collecting threadpoolctl>=2.0.0  Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)  Collecting python-dateutil>=2.8.2  Using cached python\_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)  Collecting pytz>=2020.1  Using cached pytz-2023.3-py2.py3-none-any.whl (502 kB)  Collecting six>=1.5  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)  Installing collected packages: pytz, tzdata, threadpoolctl, six, numpy, joblib, scipy, python-dateutil, scikit-learn, pandas  Successfully installed joblib-1.2.0 numpy-1.24.3 pandas-2.0.1 python-dateutil-2.8.2 pytz-2023.3 scikit-learn-1.2.2 scipy-1.10.1 six-1.16.0 threadpoolctl-3.1.0 tzdata-2023.3  scipy-1.10.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (34.4 MB)  Collecting joblib>=1.1.1  Using cached joblib-1.2.0-py3-none-any.whl (297 kB)  Collecting threadpoolctl>=2.0.0  Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)  Requirement already satisfied: six>=1.5 in /root/anaconda3/envs/linear\_reg/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas==2.0.1) (1.16.0)  Installing collected packages: tzdata, threadpoolctl, numpy, joblib, scipy, pandas, scikit-learn  Successfully installed joblib-1.2.0 numpy-1.24.3 pandas-2.0.1 scikit-learn-1.2.2 scipy-1.10.1 threadpoolctl-3.1.0 tzdata-2023.3 |
| --- |

### 

## 2.2. Sinh tập dữ liệu

**Mô tả:**

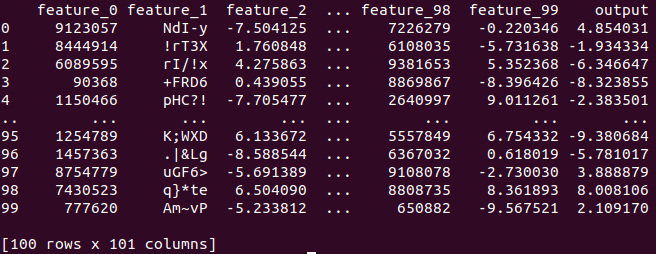
Bộ dữ liệu được sinh tự động dựa trên số mẫu và số đặc trưng của từng mẫu do người dùng tùy ý truyền vào với tùy mục đích để đánh giá hiệu năng mô hình. Các kiểu dữ liệu của từng đặc trưng được random ngẫu nhiên với các kiểu: "int", "float".

Người dùng có thể tùy chỉnh kiểu dữ liệu của các đặc trưng, số mẫu cũng như số đặc trưng của từng mẫu khi khởi tạo bộ dữ liệu. Bộ dữ liệu được lưu dưới dạng .csv sau khi khởi tạo với tên và đường dẫn tùy theo người dùng định nghĩa.

**Triển khai code tạo dữ liệu:**

| def make\_data(NUM\_SAMPLES, NUM\_FEATURES):  X, y = make\_regression(n\_samples=NUM\_SAMPLES, n\_features=NUM\_FEATURES,  n\_informative=NUM\_FEATURES, noise=0.5)  data = pd.DataFrame(X, columns=['feature' + str(i) for i in range(1, NUM\_FEATURES + 1)], dtype=np.float16)  data['output'] = np.array(y, dtype=np.float16)  return data  data = make\_data(100, 100) |
| --- |

**Kết quả:**

****

## 2.3. Xử lý tập dữ liệu

Chia dữ liệu thành 2 tập train test với cùng kích thước với mục tiêu đánh giá performance lúc train và inference model với cùng kích thước bộ dữ liệu. Vì trong phần này tôi chưa tập trung vào kết quả train của mô hình nên không áp dụng các phương pháp tiền xử lý dữ liệu như data normalization và data standardization.

| def test\_train(data, NUM\_SAMPLES):  X\_train, y\_train = data.iloc[:int(NUM\_SAMPLES / 2)].drop(['output'], axis=1), data.iloc[:int(NUM\_SAMPLES / 2)]['output']  X\_test, y\_test = data.iloc[int(NUM\_SAMPLES / 2):].drop(['output'], axis=1), data.iloc[int(NUM\_SAMPLES / 2):]['output']  return (X\_train, y\_train, X\_test, y\_test) |
| --- |

## 2.4. Huấn luyện các mô hình

### 2.4.1 Quá trình huấn luyện mô hình Linear Regression

| def fitting(X\_train, y\_train):  lm = LinearRegression(n\_jobs=1)  lm.fit(X\_train, y\_train)  del X\_train  del y\_train  return lm |
| --- |

### 2.4.2 Quá trình huấn luyện mô hình Logistic Regression

| def fitting(X\_train, y\_train):  lm = LogisticRegression(n\_jobs=1)  lm.fit(X\_train, y\_train)  del X\_train  del y\_train  return lm |
| --- |

### 2.4.3 Quá trình huấn luyện mô hình SVM

| def fitting(X\_train, y\_train):  lm = SVR(C=1.0, epsilon=0.2)  lm.fit(X\_train, y\_train)  del X\_train  del y\_train  return lm |
| --- |

# III. Build nhóm mô hình Deep learning

## 3.1 Build mô hình RecVAE

### 3.1.1. Cài đặt môi trường

#### Tạo môi trường ảo

| $ conda create -n RecVAE --clone root  Source: /root/anaconda3  Destination: /root/anaconda3/envs/RecVAE  The following packages cannot be cloned out of the root environment:  - defaults/linux-64::conda-23.3.1-py310h06a4308\_0  - defaults/linux-64::conda-build-3.24.0-py310h06a4308\_0  - defaults/noarch::conda-token-0.4.0-pyhd3eb1b0\_0  - defaults/linux-64::navigator-updater-0.3.0-py310h06a4308\_0  - defaults/linux-64::anaconda-navigator-2.4.0-py310h06a4308\_0  Packages: 139  Files: 91  Downloading and Extracting Packages  Downloading and Extracting Packages  Preparing transaction: done  Verifying transaction: done  Executing transaction: done  #  # To activate this environment, use  #  # $ conda activate RecVAE  #  # To deactivate an active environment, use  #  # $ conda deactivate |
| --- |

#### Setup các thư viện

Cài đặt torch==1.12.1+cpu qua file .whl tải từ https://download.pytorch.org/whl/torch/ :

| $ pip install torch-1.12.0+cpu-cp310-cp310-linux\_x86\_64.whl  Processing ./torch-1.12.0+cpu-cp310-cp310-linux\_x86\_64.whl  Collecting typing-extensions  Using cached typing\_extensions-4.5.0-py3-none-any.whl (27 kB)  Installing collected packages: typing-extensions, torch  Successfully installed torch-1.12.0+cpu typing-extensions-4.5.0 |
| --- |

Cài đặt các gói thư viện khác

#### 

| $ pip install scipy pandas tqdm  Collecting scipy  Using cached scipy-1.10.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (34.4 MB)  Collecting pandas  Using cached pandas-2.0.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (12.3 MB)  Requirement already satisfied: tqdm in /root/anaconda3/envs/RecVAE/lib/python3.10/site-packages (4.65.0)  Collecting numpy<1.27.0,>=1.19.5  Using cached numpy-1.24.3-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (17.3 MB)  Requirement already satisfied: python-dateutil>=2.8.2 in /root/anaconda3/envs/RecVAE/lib/python3.10/site-packages (from pandas) (2.8.2)  Requirement already satisfied: pytz>=2020.1 in /root/anaconda3/envs/RecVAE/lib/python3.10/site-packages (from pandas) (2022.7)  Collecting tzdata>=2022.1  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)  Requirement already satisfied: six>=1.5 in /root/anaconda3/envs/RecVAE/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)  Installing collected packages: tzdata, numpy, scipy, pandas  Successfully installed numpy-1.24.3 pandas-2.0.1 scipy-1.10.1 tzdata-2023.3 |
| --- |

### 3.1.2. Sinh dữ liệu

#### Do mô hình sử dụng input trực tiếp vào model sau khi qua xử lý dữ liệu là sparse matrix, số feature của input tỷ lệ thuận với max(user), max(item) nên chúng ta có thể sử dụng tập movielens với nhiều tập dữ liệu có số user, item lớn khác nhau như:

#### MovieLens 1M Dataset: 1 million sample, from 6000 users on 4000 movies.

#### MovieLens 10M Dataset: 10 million sample, 100,000 tag, 10,000 movies , 72,000 users

#### MovieLens 20M Dataset: 20 million ratings and 465,000 tag, 27,000 movies, 138,000 users

### 3.1.3. Xử lý tập dữ liệu

| dataset = args.dataset  output\_dir = args.output\_dir  threshold = args.threshold  min\_uc = args.min\_items\_per\_user  min\_sc = args.min\_users\_per\_item  n\_heldout\_users = args.heldout\_users  raw\_data = pd.read\_csv(dataset, header=0)  raw\_data = raw\_data[raw\_data['rating'] > threshold]  raw\_data.head()  def get\_count(tp, id):  playcount\_groupbyid = tp[[id]].groupby(id, as\_index=False)  count = playcount\_groupbyid.size()  return count  # def filter\_triplets(tp, min\_uc):  # usercount = get\_count(tp, 'user\_id')  # if min\_uc > 0:  # print('min\_uc', min\_uc)  # print(usercount.loc[usercount['size'] >= 5, 'user\_id'])  # tp = tp[tp['user\_id'].isin(usercount.loc[usercount['size'] >= min\_uc, 'user\_id'])]  # # Update both usercount and itemcount after filtering  # usercount, itemcount = get\_count(tp, 'user\_id'), get\_count(tp, 'content\_id')  # return tp, usercount, itemcount  def filter\_triplets(tp, min\_uc=min\_uc, min\_sc=min\_sc):  print(min\_uc, min\_sc)  if min\_sc > 0:  itemcount = get\_count(tp, 'movieId')  # print('itemcount:',itemcount.head())  # print()  # print(itemcount.index[itemcount['size'] >= min\_sc])  tp = tp[tp['movieId'].isin(itemcount.index[itemcount['size'] >= min\_sc])]    if min\_uc > 0:  usercount = get\_count(tp, 'userId')  print("usercount", usercount)  tp = tp[tp['userId'].isin(usercount.index[usercount['size'] >= min\_uc])]    usercount, itemcount = get\_count(tp, 'userId'), get\_count(tp, 'movieId')  return tp, usercount, itemcount  raw\_data, user\_activity, item\_popularity = filter\_triplets(raw\_data)  sparsity = 1. \* raw\_data.shape[0] / (user\_activity.shape[0] \* item\_popularity.shape[0])  print("After filtering, there are %d watching events from %d users and %d movies (sparsity: %.3f%%)" %  (raw\_data.shape[0], user\_activity.shape[0], item\_popularity.shape[0], sparsity \* 100))  unique\_uid = user\_activity.index  np.random.seed(98765)  idx\_perm = np.random.permutation(unique\_uid.size)  unique\_uid = unique\_uid[idx\_perm]  n\_users = unique\_uid.size  tr\_users = unique\_uid[:(n\_users - n\_heldout\_users \* 2)]  vd\_users = unique\_uid[(n\_users - n\_heldout\_users \* 2): (n\_users - n\_heldout\_users)]  te\_users = unique\_uid[(n\_users - n\_heldout\_users):]  train\_plays = raw\_data.loc[raw\_data['userId'].isin(tr\_users)]  unique\_sid = pd.unique(train\_plays['movieId'])  show2id = dict((sid, i) for (i, sid) in enumerate(unique\_sid))  profile2id = dict((pid, i) for (i, pid) in enumerate(unique\_uid))  if not os.path.exists(output\_dir):  os.makedirs(output\_dir)  with open(os.path.join(output\_dir, 'unique\_sid.txt'), 'w') as f:  for sid in unique\_sid:  f.write('%s\n' % sid)    with open(os.path.join(output\_dir, 'unique\_uid.txt'), 'w') as f:  for uid in unique\_uid:  f.write('%s\n' % uid)  def split\_train\_test\_proportion(data, test\_prop=0.2):  data\_grouped\_by\_user = data.groupby('userId')  tr\_list, te\_list = list(), list()  np.random.seed(98765)  for i, (\_, group) in enumerate(data\_grouped\_by\_user):  n\_items\_u = len(group)  if n\_items\_u >= 5:  idx = np.zeros(n\_items\_u, dtype='bool')  idx[np.random.choice(n\_items\_u, size=int(test\_prop \* n\_items\_u), replace=False).astype('int64')] = True  tr\_list.append(group[np.logical\_not(idx)])  te\_list.append(group[idx])  else:  tr\_list.append(group)  if i % 1000 == 0:  print("%d users sampled" % i)  sys.stdout.flush()  data\_tr = pd.concat(tr\_list)  data\_te = pd.concat(te\_list)    return data\_tr, data\_te  vad\_plays = raw\_data.loc[raw\_data['userId'].isin(vd\_users)]  vad\_plays = vad\_plays.loc[vad\_plays['movieId'].isin(unique\_sid)]  vad\_plays\_tr, vad\_plays\_te = split\_train\_test\_proportion(vad\_plays)  test\_plays = raw\_data.loc[raw\_data['userId'].isin(te\_users)]  test\_plays = test\_plays.loc[test\_plays['movieId'].isin(unique\_sid)]  test\_plays\_tr, test\_plays\_te = split\_train\_test\_proportion(test\_plays)  def numerize(tp):  uid = list(map(lambda x: profile2id[x], tp['userId']))  sid = list(map(lambda x: show2id[x], tp['movieId']))  return pd.DataFrame(data={'uid': uid, 'sid': sid}, columns=['uid', 'sid'])  train\_data = numerize(train\_plays)  train\_data.to\_csv(os.path.join(output\_dir, 'train.csv'), index=False)  vad\_data\_tr = numerize(vad\_plays\_tr)  vad\_data\_tr.to\_csv(os.path.join(output\_dir, 'validation\_tr.csv'), index=False)  vad\_data\_te = numerize(vad\_plays\_te)  vad\_data\_te.to\_csv(os.path.join(output\_dir, 'validation\_te.csv'), index=False)  test\_data\_tr = numerize(test\_plays\_tr)  test\_data\_tr.to\_csv(os.path.join(output\_dir, 'test\_tr.csv'), index=False)  test\_data\_te = numerize(test\_plays\_te)  test\_data\_te.to\_csv(os.path.join(output\_dir, 'test\_te.csv'), index=False) |
| --- |

### 3.1.4. Quá trình huấn luyện mô hình

Chúng tôi thử nghiệm huấn luyện mô hình trên 3 bộ dữ liệu là movies len 1m, 10m và 20m, mỗi lần mô hình huấn luyện chạy trên 10 epoch cùng với load data với batch\_size = 512.

* Qúa trình huấn luyện trên bộ dữ liệu movieslen-1m:

#### 

| [54]2023-05-21 10:50:01,894:  >>>>> Main Config  [Dataset]  data\_dir: data/recsys\_data  dataset: ml-1m  min\_user\_per\_item: 1  min\_item\_per\_user: 10  leave\_k: 5  popularity\_order: True  [Evaluator]  ks: [20, 50]  [EarlyStop]  early\_stop: 50  early\_stop\_measure: Recall@50  [Experiment]  num\_epochs: 10  verbose: 0  print\_step: 1  test\_step: 1  test\_from: 1  model\_name: RecVAE  seed: 2023  gpu: 0  >>>>> model Config  [Model]  enc\_dims: 300  dropout: 0.5  total\_anneal\_steps: 200000  batch\_size: 512  test\_batch\_size: 512  lr: 0.001  act: tanh  latent\_dim: 200  [54]2023-05-21 10:50:01,895:  Dataset: ml-1m  # of users: 6040  # of items: 3706 |
| --- |

#### Kết quả

| * [54]2023-05-21 10:50:02,668: train start ... ! * [54]2023-05-21 10:50:05,511: epoch= 1, loss=15569.824, train time=1.72, epoch time=2.68 (1.72 + 0.96), Prec@20=0.0118, Prec@50=0.0100, Recall@20=0.0473, Recall@50=0.1000, NDCG@20=0.0282, NDCG@50=0.0458 * [54]2023-05-21 10:50:08,236: epoch= 2, loss=14768.546, train time=1.72, epoch time=2.72 (1.72 + 1.00), Prec@20=0.0085, Prec@50=0.0069, Recall@20=0.0341, Recall@50=0.0688, NDCG@20=0.0212, NDCG@50=0.0328 * [54]2023-05-21 10:50:11,091: epoch= 3, loss=14387.250, train time=1.82, epoch time=2.85 (1.82 + 1.03), Prec@20=0.0090, Prec@50=0.0080, Recall@20=0.0358, Recall@50=0.0805, NDCG@20=0.0220, NDCG@50=0.0368 * [54]2023-05-21 10:50:13,845: epoch= 4, loss=14299.906, train time=1.75, epoch time=2.75 (1.75 + 1.00), Prec@20=0.0116, Prec@50=0.0098, Recall@20=0.0462, Recall@50=0.0977, NDCG@20=0.0267, NDCG@50=0.0439 * [54]2023-05-21 10:50:16,577: epoch= 5, loss=14195.839, train time=1.74, epoch time=2.73 (1.74 + 0.99), Prec@20=0.0104, Prec@50=0.0097, Recall@20=0.0418, Recall@50=0.0973, NDCG@20=0.0252, NDCG@50=0.0437 * [54]2023-05-21 10:50:19,471: epoch= 6, loss=14105.813, train time=1.77, epoch time=2.89 (1.77 + 1.12), Prec@20=0.0119, Prec@50=0.0105, Recall@20=0.0476, Recall@50=0.1047, NDCG@20=0.0282, NDCG@50=0.0472 * [54]2023-05-21 10:50:22,302: epoch= 7, loss=13917.801, train time=1.81, epoch time=2.83 (1.81 + 1.02), Prec@20=0.0115, Prec@50=0.0108, Recall@20=0.0459, Recall@50=0.1082, NDCG@20=0.0277, NDCG@50=0.0485 * [54]2023-05-21 10:50:25,207: epoch= 8, loss=14011.608, train time=1.89, epoch time=2.90 (1.89 + 1.02), Prec@20=0.0132, Prec@50=0.0121, Recall@20=0.0526, Recall@50=0.1206, NDCG@20=0.0325, NDCG@50=0.0551 * [54]2023-05-21 10:50:28,042: epoch= 9, loss=13953.657, train time=1.83, epoch time=2.83 (1.83 + 1.00), Prec@20=0.0122, Prec@50=0.0114, Recall@20=0.0489, Recall@50=0.1141, NDCG@20=0.0301, NDCG@50=0.0516 * [54]2023-05-21 10:50:30,955: epoch= 10, loss=13732.756, train time=1.81, epoch time=2.91 (1.81 + 1.09), Prec@20=0.0129, Prec@50=0.0122, Recall@20=0.0517, Recall@50=0.1220, NDCG@20=0.0304, NDCG@50=0.0538 |
| --- |

* Quá trình huấn luyện trên tập dữ liệu movieslen-10m:

#### 

| [54]2023-05-21 10:51:18,988:  >>>>> Main Config  [Dataset]  data\_dir: data/recsys\_data  dataset: ml-10m  min\_user\_per\_item: 1  min\_item\_per\_user: 10  leave\_k: 5  popularity\_order: True  [Evaluator]  ks: [20, 50]  [EarlyStop]  early\_stop: 50  early\_stop\_measure: Recall@50  [Experiment]  num\_epochs: 10  verbose: 0  print\_step: 1  test\_step: 1  test\_from: 1  model\_name: RecVAE  seed: 2023  gpu: 0  >>>>> model Config  [Model]  enc\_dims: 300  dropout: 0.5  total\_anneal\_steps: 200000  batch\_size: 512  test\_batch\_size: 512  lr: 0.001  act: tanh  latent\_dim: 200  [54]2023-05-21 10:51:18,989:  Dataset: ml-10m  # of users: 69878  # of items: 10677 |
| --- |

#### 

| * [54]2023-05-21 10:51:19,813: train start ... ! * [54]2023-05-21 10:56:31,051: epoch= 1, loss=151369.859, train time=263.69, epoch time=305.85 (263.69 + 42.16), Prec@20=0.0106, Prec@50=0.0132, Recall@20=0.0425, Recall@50=0.1316, NDCG@20=0.0251, NDCG@50=0.0545 * [54]2023-05-21 11:06:57,812: epoch= 2, loss=144210.375, train time=581.93, epoch time=622.14 (581.93 + 40.22), Prec@20=0.0117, Prec@50=0.0143, Recall@20=0.0467, Recall@50=0.1428, NDCG@20=0.0277, NDCG@50=0.0595 * [54]2023-05-21 11:10:40,637: epoch= 3, loss=139849.359, train time=191.38, epoch time=218.56 (191.38 + 27.18), Prec@20=0.0119, Prec@50=0.0146, Recall@20=0.0478, Recall@50=0.1461, NDCG@20=0.0281, NDCG@50=0.0606 * [54]2023-05-21 11:12:50,651: epoch= 4, loss=138000.016, train time=103.58, epoch time=129.75 (103.58 + 26.17), Prec@20=0.0124, Prec@50=0.0153, Recall@20=0.0494, Recall@50=0.1526, NDCG@20=0.0289, NDCG@50=0.0630 * [54]2023-05-21 11:14:38,293: epoch= 5, loss=136201.891, train time=80.40, epoch time=107.47 (80.40 + 27.07), Prec@20=0.0124, Prec@50=0.0158, Recall@20=0.0497, Recall@50=0.1577, NDCG@20=0.0289, NDCG@50=0.0645 * [54]2023-05-21 11:15:55,324: epoch= 6, loss=135077.734, train time=48.99, epoch time=76.88 (48.99 + 27.90), Prec@20=0.0123, Prec@50=0.0159, Recall@20=0.0490, Recall@50=0.1590, NDCG@20=0.0280, NDCG@50=0.0643 * [54]2023-05-21 11:17:11,783: epoch= 7, loss=134139.047, train time=48.25, epoch time=76.37 (48.25 + 28.12), Prec@20=0.0121, Prec@50=0.0161, Recall@20=0.0485, Recall@50=0.1605, NDCG@20=0.0274, NDCG@50=0.0644 * [54]2023-05-21 11:18:27,437: epoch= 8, loss=133358.734, train time=47.63, epoch time=75.55 (47.63 + 27.92), Prec@20=0.0120, Prec@50=0.0161, Recall@20=0.0480, Recall@50=0.1614, NDCG@20=0.0268, NDCG@50=0.0642 * [54]2023-05-21 11:20:19,404: epoch= 9, loss=132620.109, train time=83.16, epoch time=111.88 (83.16 + 28.71), Prec@20=0.0115, Prec@50=0.0161, Recall@20=0.0461, Recall@50=0.1608, NDCG@20=0.0255, NDCG@50=0.0634 * [54]2023-05-21 11:22:53,069: epoch= 10, loss=132040.609, train time=123.69, epoch time=152.81 (123.69 + 29.11), Prec@20=0.0111, Prec@50=0.0160, Recall@20=0.0443, Recall@50=0.1600, NDCG@20=0.0241, NDCG@50=0.0623 |
| --- |

#### 

|  |
| --- |

* Huấn luyện trên tập dữ liệu movieslen-20m

Kích thước của bộ dữ liệu này rất lớn không thể xử lý được trên thiết bị phần cứng được cung cấp.

## 3.2 Build mô hình BERT4REC

### 3.2.1. Cài đặt môi trường

#### 3.2.1.1. Tạo môi trường ảo

| $ conda create -n bert4rec  Collecting package metadata (current\_repodata.json): done  Solving environment: done  ## Package Plan ##  environment location: /root/anaconda3/envs/bert4rec  Proceed ([y]/n)? y  Preparing transaction: done  Verifying transaction: done  Executing transaction: done  #  # To activate this environment, use  #  # $ conda activate bert4rec  #  # To deactivate an active environment, use  #  # $ conda deactivate |
| --- |

Kích hoạt môi trường vừa mới khởi tạo:

| $ conda activate bert4rec |
| --- |

#### 3.2.1.2. Setup các thư viện

Cài đặt tensorflow 1.12.0

| $ pip install tensorflow==1.12.0  DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support  Requirement already satisfied: pip in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (19.3.1)  Collecting install  Downloading https://files.pythonhosted.org/packages/45/41/7f917b56e936b749ed985dec682ebefdb039a8a9458221c8c5f0ce4f970d/install-1.3.0.tar.gz  Collecting tensorflow==1.12.0  Downloading https://files.pythonhosted.org/packages/bd/68/ec26b2cb070a5760707ec8d9491a24e5be72f4885f265bb04abf70c0f9f1/tensorflow-1.12.0-cp27-cp27mu-manylinux1\_x86\_64.whl (83.1MB)  |████████████████████████████████| 83.1MB 109kB/s  Collecting keras-preprocessing>=1.0.5  Using cached https://files.pythonhosted.org/packages/79/4c/7c3275a01e12ef9368a892926ab932b33bb13d55794881e3573482b378a7/Keras\_Preprocessing-1.1.2-py2.py3-none-any.whl  Collecting enum34>=1.1.6  Downloading https://files.pythonhosted.org/packages/6f/2c/a9386903ece2ea85e9807e0e062174dc26fdce8b05f216d00491be29fad5/enum34-1.1.10-py2-none-any.whl  Collecting astor>=0.6.0  Using cached https://files.pythonhosted.org/packages/c3/88/97eef84f48fa04fbd6750e62dcceafba6c63c81b7ac1420856c8dcc0a3f9/astor-0.8.1-py2.py3-none-any.whl  Collecting backports.weakref>=1.0rc1  Downloading https://files.pythonhosted.org/packages/88/ec/f598b633c3d5ffe267aaada57d961c94fdfa183c5c3ebda2b6d151943db6/backports.weakref-1.0.post1-py2.py3-none-any.whl  Requirement already satisfied: wheel in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from tensorflow==1.12.0) (0.37.1)  Collecting mock>=2.0.0  Downloading https://files.pythonhosted.org/packages/05/d2/f94e68be6b17f46d2c353564da56e6fb89ef09faeeff3313a046cb810ca9/mock-3.0.5-py2.py3-none-any.whl  Collecting tensorboard<1.13.0,>=1.12.0  Downloading https://files.pythonhosted.org/packages/51/ae/9840c4837c6f54034ac942b5344396e8c3d74686a9bd29beafdf633cc221/tensorboard-1.12.2-py2-none-any.whl (3.0MB)  |████████████████████████████████| 3.1MB 2.4MB/s  Collecting gast>=0.2.0  Downloading https://files.pythonhosted.org/packages/e4/41/f26f62ebef1a80148e20951a6e9ef4d0ebbe2090124bc143da26e12a934c/gast-0.5.4.tar.gz  Collecting termcolor>=1.1.0  Downloading https://files.pythonhosted.org/packages/8a/48/a76be51647d0eb9f10e2a4511bf3ffb8cc1e6b14e9e4fab46173aa79f981/termcolor-1.1.0.tar.gz  Collecting protobuf>=3.6.1  Downloading https://files.pythonhosted.org/packages/e9/06/5606088c9fbfb924d0f228350e7ec8707fe8f8a6f3a024ace4f0da81d9ce/protobuf-3.17.3-cp27-cp27mu-manylinux\_2\_5\_x86\_64.manylinux1\_x86\_64.whl (1.0MB)  |████████████████████████████████| 1.0MB 2.7MB/s  Collecting absl-py>=0.1.6  Downloading https://files.pythonhosted.org/packages/25/d9/22a0b010487da88200c3f0672c67e892b5399b7a13d105d3863511399f6e/absl-py-0.15.0.tar.gz (114kB)  |████████████████████████████████| 122kB 3.2MB/s  Collecting six>=1.10.0  Using cached https://files.pythonhosted.org/packages/d9/5a/e7c31adbe875f2abbb91bd84cf2dc52d792b5a01506781dbcf25c91daf11/six-1.16.0-py2.py3-none-any.whl  Collecting keras-applications>=1.0.6  Downloading https://files.pythonhosted.org/packages/21/56/4bcec5a8d9503a87e58e814c4e32ac2b32c37c685672c30bc8c54c6e478a/Keras\_Applications-1.0.8.tar.gz (289kB)  |████████████████████████████████| 296kB 3.2MB/s  Collecting grpcio>=1.8.6  Downloading https://files.pythonhosted.org/packages/bd/81/6c704c002a992b9d6466c739e3e7687e0bb2365d8cd63d7fc8e95d502cb6/grpcio-1.41.1.tar.gz (21.2MB)  |████████████████████████████████| 21.2MB 4.7MB/s  Collecting numpy>=1.13.3  Downloading https://files.pythonhosted.org/packages/3a/5f/47e578b3ae79e2624e205445ab77a1848acdaa2929a00eeef6b16eaaeb20/numpy-1.16.6-cp27-cp27mu-manylinux1\_x86\_64.whl (17.0MB)  |████████████████████████████████| 17.0MB 103kB/s  Collecting funcsigs>=1; python\_version < "3.3"  Downloading https://files.pythonhosted.org/packages/69/cb/f5be453359271714c01b9bd06126eaf2e368f1fddfff30818754b5ac2328/funcsigs-1.0.2-py2.py3-none-any.whl  Collecting futures>=3.1.1; python\_version < "3"  Downloading https://files.pythonhosted.org/packages/d4/ea/9d513529a89bcbcd07c8acbac9eecfad29e7562e0b9d69d14f475987ad70/futures-3.4.0-py2-none-any.whl  Collecting markdown>=2.6.8  Downloading https://files.pythonhosted.org/packages/c0/4e/fd492e91abdc2d2fcb70ef453064d980688762079397f779758e055f6575/Markdown-3.1.1-py2.py3-none-any.whl (87kB)  |████████████████████████████████| 92kB 2.7MB/s  Collecting werkzeug>=0.11.10  Downloading https://files.pythonhosted.org/packages/cc/94/5f7079a0e00bd6863ef8f1da638721e9da21e5bacee597595b318f71d62e/Werkzeug-1.0.1-py2.py3-none-any.whl (298kB)  |████████████████████████████████| 307kB 4.6MB/s  Collecting h5py  Downloading https://files.pythonhosted.org/packages/12/90/3216b8f6d69905a320352a9ca6802a8e39fdb1cd93133c3d4163db8d5f19/h5py-2.10.0-cp27-cp27mu-manylinux1\_x86\_64.whl (2.8MB)  |████████████████████████████████| 2.8MB 4.8MB/s  Requirement already satisfied: setuptools>=36 in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from markdown>=2.6.8->tensorboard<1.13.0,>=1.12.0->tensorflow==1.12.0) (44.0.0.post20200106)  Building wheels for collected packages: install, gast, termcolor, absl-py, keras-applications, grpcio  Building wheel for install (setup.py) ... done  Created wheel for install: filename=install-1.3.0-cp27-none-any.whl size=2080 sha256=f0b45627adeef6f82d1895121f3e9fcbe156ea6a5e5c33d279d87bf0e817921f  Stored in directory: /root/.cache/pip/wheels/41/70/48/0c5542b7de04d4490d910cdf760f29e1ee0beb8a6e1d3bf1f2  Building wheel for gast (setup.py) ... done  Created wheel for gast: filename=gast-0.5.4-cp27-none-any.whl size=19674 sha256=66af0d93003cdde18c526707a41958fcdd22a395081c9b4363b4b23cfb619b26  Stored in directory: /root/.cache/pip/wheels/5c/db/60/54f7f419f7ecd013132e39ced59187b011ba409ffb84c0f23e  Building wheel for termcolor (setup.py) ... done  Created wheel for termcolor: filename=termcolor-1.1.0-cp27-none-any.whl size=4833 sha256=c94045005d533e1c934df5c0830d46e7eb728de689d5a43b0d798bdd303fecb2  Stored in directory: /root/.cache/pip/wheels/7c/06/54/bc84598ba1daf8f970247f550b175aaaee85f68b4b0c5ab2c6  Building wheel for absl-py (setup.py) ... done  Created wheel for absl-py: filename=absl\_py-0.15.0-cp27-none-any.whl size=132010 sha256=ab9f556fda00bccb52860d2422a3f7ed66f2f73bd649e77e8313284dcb60c5b5  Stored in directory: /root/.cache/pip/wheels/19/f3/f9/f5b598f0b550f4140026ca6a7aa6b016929cefc369b5104df6  Building wheel for keras-applications (setup.py) ... done  Created wheel for keras-applications: filename=Keras\_Applications-1.0.8-cp27-none-any.whl size=50703 sha256=ada90013c909013ff59ff4ede2f7554db82de8a720c0127c347f26ee3d618b10  Stored in directory: /root/.cache/pip/wheels/dd/f2/5d/2689b5547f32c4e258c3b7ccbe7f1d0f2afbb84fb01e830792  Building wheel for grpcio (setup.py) ... error  Running setup.py clean for grpcio  Successfully built install gast termcolor absl-py keras-applications  Failed to build grpcio  Installing collected packages: install, six, numpy, keras-preprocessing, enum34, astor, backports.weakref, funcsigs, mock, protobuf, futures, grpcio, markdown, werkzeug, tensorboard, gast, termcolor, absl-py, h5py, keras-applications, tensorflow |
| --- |

### 

### Cài đặt grpcio

| $ pip install grpcio==1.36.1  DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support  Collecting grpcio==1.36.1  Downloading https://files.pythonhosted.org/packages/46/43/27d993a138989c7c7e758836780a92dab37dc71ecd1de57fc0ae2cc97727/grpcio-1.36.1-cp27-cp27mu-manylinux2010\_x86\_64.whl (3.9MB)  |████████████████████████████████| 3.9MB 3.1MB/s  Requirement already satisfied: enum34>=1.0.4; python\_version < "3.4" in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from grpcio==1.36.1) (1.1.10)  Requirement already satisfied: futures>=2.2.0; python\_version < "3.2" in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from grpcio==1.36.1) (3.4.0)  Requirement already satisfied: six>=1.5.2 in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from grpcio==1.36.1) (1.16.0)  Installing collected packages: grpcio  Successfully installed grpcio-1.36.1 |
| --- |

#### 

### Cài đặt các thư viện khác

| $ pip install pandas scipy  DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support  Collecting pandas  Downloading https://files.pythonhosted.org/packages/db/83/7d4008ffc2988066ff37f6a0bb6d7b60822367dcb36ba5e39aa7801fda54/pandas-0.24.2-cp27-cp27mu-manylinux1\_x86\_64.whl (10.1MB)  |████████████████████████████████| 10.1MB 787kB/s  Collecting scipy  Downloading https://files.pythonhosted.org/packages/24/40/11b12af7f322c1e20446c037c47344d89bab4922b8859419d82cf56d796d/scipy-1.2.3-cp27-cp27mu-manylinux1\_x86\_64.whl (24.8MB)  |████████████████████████████████| 24.8MB 5.5MB/s  Collecting pytz>=2011k  Using cached https://files.pythonhosted.org/packages/7f/99/ad6bd37e748257dd70d6f85d916cafe79c0b0f5e2e95b11f7fbc82bf3110/pytz-2023.3-py2.py3-none-any.whl  Requirement already satisfied: numpy>=1.12.0 in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from pandas) (1.16.6)  Collecting python-dateutil>=2.5.0  Using cached https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9b62bbb257d0355c7f6128853c78955f57342a56d/python\_dateutil-2.8.2-py2.py3-none-any.whl  Requirement already satisfied: six>=1.5 in /root/anaconda3/envs/bert4rec/lib/python2.7/site-packages (from python-dateutil>=2.5.0->pandas) (1.16.0)  Installing collected packages: pytz, python-dateutil, pandas, scipy  Successfully installed pandas-0.24.2 python-dateutil-2.8.2 pytz-2023.3 scipy-1.2.3 |
| --- |

#### 

### 3.2.2. Sinh tập dữ liệu

#### Do mô hình sử dụng input trực tiếp vào model sau khi qua xử lý dữ liệu là sparse matrix, số feature của input tỷ lệ thuận với max(user), max(item) nên chúng ta có thể sử dụng tập movielens với nhiều tập dữ liệu có số user, item lớn khác nhau như:

#### MovieLens 100K Dataset : 100,000 sample, 1000 users, 1700 movies)

#### MovieLens 1M Dataset: 1 million sample, from 6000 users on 4000 movies

#### MovieLens 10M Dataset: 10 million sample,100,000 tag applications, 10,000 movies , 72,000 users

#### MovieLens 20M Dataset: 20 million ratings and 465,000 tag applications, 27,000 movies, 138,000 users

### 3.2.3. Xử lý tập dữ liệu

| tf.logging.set\_verbosity(tf.logging.DEBUG)  max\_seq\_length = FLAGS.max\_seq\_length  max\_predictions\_per\_seq = FLAGS.max\_predictions\_per\_seq  masked\_lm\_prob = FLAGS.masked\_lm\_prob  mask\_prob = FLAGS.mask\_prob  dupe\_factor = FLAGS.dupe\_factor  prop\_sliding\_window = FLAGS.prop\_sliding\_window  pool\_size = FLAGS.pool\_size  output\_dir = FLAGS.data\_dir  dataset\_name = FLAGS.dataset\_name  version\_id = FLAGS.signature  print  version\_id  if not os.path.isdir(output\_dir):  print(output\_dir + ' is not exist')  print(os.getcwd())  exit(1)  dataset = data\_partition(output\_dir + dataset\_name + '.txt')  [user\_train, user\_valid, user\_test, usernum, itemnum] = dataset  cc = 0.0  max\_len = 0  min\_len = 100000  for u in user\_train:  cc += len(user\_train[u])  max\_len = max(len(user\_train[u]), max\_len)  min\_len = min(len(user\_train[u]), min\_len)  print('average sequence length: %.2f' % (cc / len(user\_train)))  print('max:{}, min:{}'.format(max\_len, min\_len))  print('len\_train:{}, len\_valid:{}, len\_test:{}, usernum:{}, itemnum:{}'.  format(  len(user\_train),  len(user\_valid), len(user\_test), usernum, itemnum))  for idx, u in enumerate(user\_train):  if idx < 10:  print(user\_train[u])  print(user\_valid[u])  print(user\_test[u])  # put validate into train  for u in user\_train:  if u in user\_valid:  user\_train[u].extend(user\_valid[u])  # get the max index of the data  user\_train\_data = {  'user\_' + str(k): ['item\_' + str(item) for item in v]  for k, v in user\_train.items() if len(v) > 0  }  user\_test\_data = {  'user\_' + str(u):  ['item\_' + str(item) for item in (user\_train[u] + user\_test[u])]  for u in user\_train if len(user\_train[u]) > 0 and len(user\_test[u]) > 0  }  rng = random.Random(random\_seed)  vocab = FreqVocab(user\_test\_data)  user\_test\_data\_output = {  k: [vocab.convert\_tokens\_to\_ids(v)]  for k, v in user\_test\_data.items()  }  print('begin to generate train')  output\_filename = output\_dir + dataset\_name + version\_id + '.train.tfrecord'  gen\_samples(  user\_train\_data,  output\_filename,  rng,  vocab,  max\_seq\_length,  dupe\_factor,  short\_seq\_prob,  mask\_prob,  masked\_lm\_prob,  max\_predictions\_per\_seq,  prop\_sliding\_window,  pool\_size,  force\_last=False)  print('train:{}'.format(output\_filename))  print('begin to generate test')  output\_filename = output\_dir + dataset\_name + version\_id + '.test.tfrecord'  gen\_samples(  user\_test\_data,  output\_filename,  rng,  vocab,  max\_seq\_length,  dupe\_factor,  short\_seq\_prob,  mask\_prob,  masked\_lm\_prob,  max\_predictions\_per\_seq,  -1.0,  pool\_size,  force\_last=True)  print('test:{}'.format(output\_filename))  print('vocab\_size:{}, user\_size:{}, item\_size:{}, item\_with\_other\_size:{}'.  format(vocab.get\_vocab\_size(),  vocab.get\_user\_count(),  vocab.get\_item\_count(),  vocab.get\_item\_count() + vocab.get\_special\_token\_count()))  vocab\_file\_name = output\_dir + dataset\_name + version\_id + '.vocab'  print('vocab pickle file: ' + vocab\_file\_name)  with open(vocab\_file\_name, 'wb') as output\_file:  pickle.dump(vocab, output\_file, protocol=2)  his\_file\_name = output\_dir + dataset\_name + version\_id + '.his'  print('test data pickle file: ' + his\_file\_name)  with open(his\_file\_name, 'wb') as output\_file:  pickle.dump(user\_test\_data\_output, output\_file, protocol=2)  print('done.') |
| --- |

### 3.2.4. Quá trình huấn luyện mô hình

| INFO:tensorflow:\*\*\* train Input Files \*\*\*  INFO:tensorflow: ./data/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200.train.tfrecord  INFO:tensorflow:\*\*\* test Input Files \*\*\*  INFO:tensorflow: ./data/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200.train.tfrecord  INFO:tensorflow:Using config: {'\_save\_checkpoints\_secs': None, '\_session\_config': allow\_soft\_placement: true  graph\_options {  rewrite\_options {  meta\_optimizer\_iterations: ONE  }  }  , '\_keep\_checkpoint\_max': 5, '\_task\_type': 'worker', '\_train\_distribute': None, '\_is\_chief': True, '\_cluster\_spec': <tensorflow.python.training.server\_lib.ClusterSpec object at 0x7f2327353f90>, '\_model\_dir': 'ckpt/BERT4Rec/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200-64', '\_protocol': None, '\_save\_checkpoints\_steps': 1000, '\_keep\_checkpoint\_every\_n\_hours': 10000, '\_service': None, '\_num\_ps\_replicas': 0, '\_tf\_random\_seed': None, '\_save\_summary\_steps': 100, '\_device\_fn': None, '\_experimental\_distribute': None, '\_num\_worker\_replicas': 1, '\_task\_id': 0, '\_log\_step\_count\_steps': 100, '\_evaluation\_master': '', '\_eval\_distribute': None, '\_global\_id\_in\_cluster': 0, '\_master': ''}  INFO:tensorflow:\*\*\*\*\* Running training \*\*\*\*\*  INFO:tensorflow: Batch size = 5120  INFO:tensorflow:Skipping training since max\_steps has already saved.  INFO:tensorflow:\*\*\*\*\* Running evaluation \*\*\*\*\*  INFO:tensorflow: Batch size = 5120  INFO:tensorflow:run init  INFO:tensorflow:Calling model\_fn.  INFO:tensorflow:\*\*\* Features \*\*\*  INFO:tensorflow: name = info, shape = (?, 1)  INFO:tensorflow: name = input\_ids, shape = (?, 200)  INFO:tensorflow: name = input\_mask, shape = (?, 200)  INFO:tensorflow: name = masked\_lm\_ids, shape = (?, 40)  INFO:tensorflow: name = masked\_lm\_positions, shape = (?, 40)  INFO:tensorflow: name = masked\_lm\_weights, shape = (?, 40)  INFO:tensorflow:\*\*\*\* Trainable Variables \*\*\*\*  INFO:tensorflow: name = bert/embeddings/word\_embeddings:0, shape = (3420, 64)  INFO:tensorflow: name = bert/embeddings/token\_type\_embeddings:0, shape = (2, 64)  INFO:tensorflow: name = bert/embeddings/position\_embeddings:0, shape = (200, 64)  INFO:tensorflow: name = bert/embeddings/LayerNorm/beta:0, shape = (64,)  INFO:tensorflow: name = bert/embeddings/LayerNorm/gamma:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/self/query/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/self/query/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/self/key/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/self/key/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/self/value/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/self/value/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/output/dense/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/output/dense/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/output/LayerNorm/beta:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/attention/output/LayerNorm/gamma:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/intermediate/dense/kernel:0, shape = (64, 256)  INFO:tensorflow: name = bert/encoder/layer\_0/intermediate/dense/bias:0, shape = (256,)  INFO:tensorflow: name = bert/encoder/layer\_0/output/dense/kernel:0, shape = (256, 64)  INFO:tensorflow: name = bert/encoder/layer\_0/output/dense/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/output/LayerNorm/beta:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_0/output/LayerNorm/gamma:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/self/query/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/self/query/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/self/key/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/self/key/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/self/value/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/self/value/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/output/dense/kernel:0, shape = (64, 64)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/output/dense/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/output/LayerNorm/beta:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/attention/output/LayerNorm/gamma:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/intermediate/dense/kernel:0, shape = (64, 256)  INFO:tensorflow: name = bert/encoder/layer\_1/intermediate/dense/bias:0, shape = (256,)  INFO:tensorflow: name = bert/encoder/layer\_1/output/dense/kernel:0, shape = (256, 64)  INFO:tensorflow: name = bert/encoder/layer\_1/output/dense/bias:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/output/LayerNorm/beta:0, shape = (64,)  INFO:tensorflow: name = bert/encoder/layer\_1/output/LayerNorm/gamma:0, shape = (64,)  INFO:tensorflow: name = cls/predictions/transform/dense/kernel:0, shape = (64, 64)  INFO:tensorflow: name = cls/predictions/transform/dense/bias:0, shape = (64,)  INFO:tensorflow: name = cls/predictions/transform/LayerNorm/beta:0, shape = (64,)  INFO:tensorflow: name = cls/predictions/transform/LayerNorm/gamma:0, shape = (64,)  INFO:tensorflow: name = cls/predictions/output\_bias:0, shape = (3420,)  INFO:tensorflow:Done calling model\_fn.  INFO:tensorflow:Starting evaluation at 2023-05-20-16:04:45  load user history from :./data/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200.his  load vocab from :./data/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200.vocab  INFO:tensorflow:Graph was finalized.  2023-05-20 16:04:46.523758: I tensorflow/core/platform/cpu\_feature\_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA  INFO:tensorflow:Restoring parameters from ckpt/BERT4Rec/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200-64/model.ckpt-10  INFO:tensorflow:Running local\_init\_op.  INFO:tensorflow:Done running local\_init\_op.  2023-05-20 16:04:47.103306: W tensorflow/core/framework/allocator.cc:122] Allocation of 2801664000 exceeds 10% of system memory.  2023-05-20 16:04:59.614104: W tensorflow/core/framework/allocator.cc:122] Allocation of 819200000 exceeds 10% of system memory.  2023-05-20 16:05:11.128600: W tensorflow/core/framework/allocator.cc:122] Allocation of 1638400000 exceeds 10% of system memory.  2023-05-20 16:05:22.012461: W tensorflow/core/framework/allocator.cc:122] Allocation of 1048576000 exceeds 10% of system memory.  2023-05-20 16:05:23.375643: W tensorflow/core/framework/allocator.cc:122] Allocation of 1048576000 exceeds 10% of system memory.  ............................................................ndcg@1:0.0100993377483, hit@1:0.0100993377483， ndcg@5:0.029370533419, hit@5:0.0503311258278, ndcg@10:0.0460476502248, hit@10:0.102483443709, ap:0.0519040799384, valid\_user:6040.0  INFO:tensorflow:Finished evaluation at 2023-05-20-16:06:27  INFO:tensorflow:Saving dict for global step 10: global\_step = 10, loss = 8.135874, masked\_lm\_accuracy = 0.0, masked\_lm\_loss = 8.134932  INFO:tensorflow:Saving 'checkpoint\_path' summary for global step 10: ckpt/BERT4Rec/ml-1m-mp1.0-sw0.5-mlp0.2-df10-mpps40-msl200-64/model.ckpt-10  INFO:tensorflow:\*\*\*\*\* Eval results \*\*\*\*\*  INFO:tensorflow:{  "attention\_probs\_dropout\_prob": 0.2,  "hidden\_act": "gelu",  "hidden\_dropout\_prob": 0.2,  "hidden\_size": 64,  "initializer\_range": 0.02,  "intermediate\_size": 256,  "max\_position\_embeddings": 200,  "num\_attention\_heads": 2,  "num\_hidden\_layers": 2,  "type\_vocab\_size": 2,  "vocab\_size": 3420  }  INFO:tensorflow: global\_step = 10  INFO:tensorflow: loss = 8.135874  INFO:tensorflow: masked\_lm\_accuracy = 0.0  INFO:tensorflow: masked\_lm\_loss = 8.134932  time training: 104.266243935 |
| --- |

## 3.3 Build mô hình CONTRASTIVE LEARNING

### 3.3.1. Cài đặt môi trường

#### 3.3.1.1. Tạo môi trường ảo

Vì cài đặt offline nên ta sử dụng lệnh "conda create -n yourenvname --clone root" để tạo môi trường mới với phiên bản python và các gói thư viện mặc định theo môi trường base được khởi tạo khi cài đặt hoàn tất anaconda3.

| $ conda create -n contrastive\_learning --clone root  Source: /root/anaconda3  Destination: /root/anaconda3/envs/linear\_reg  The following packages cannot be cloned out of the root environment:  - defaults/linux-64::conda-23.3.1-py310h06a4308\_0  - defaults/linux-64::conda-build-3.24.0-py310h06a4308\_0  - defaults/noarch::conda-token-0.4.0-pyhd3eb1b0\_0  - defaults/linux-64::navigator-updater-0.3.0-py310h06a4308\_0  - defaults/linux-64::anaconda-navigator-2.4.0-py310h06a4308\_0  Packages: 139  Files: 91  Downloading and Extracting Packages  Downloading and Extracting Packages  Preparing transaction: done  Verifying transaction: done  Executing transaction: done  #  # To activate this environment, use  #  # $ conda activate contrastive\_learning  #  # To deactivate an active environment, use  #  # $ conda deactivate |
| --- |

Kích hoạt môi trường vừa mới khởi tạo:

| $ conda activate contrastive\_learning |
| --- |

#### 3.3.1.2. Setup các thư viện

Cài đặt torch==1.12.1+cpu qua file .whl tải từ https://download.pytorch.org/whl/torch/ :

| $ pip install torch-1.12.0+cpu-cp310-cp310-linux\_x86\_64.whl  Processing ./torch-1.12.0+cpu-cp310-cp310-linux\_x86\_64.whl  Collecting typing-extensions  Using cached typing\_extensions-4.5.0-py3-none-any.whl (27 kB)  Installing collected packages: typing-extensions, torch  Successfully installed torch-1.12.0+cpu typing-extensions-4.5.0 |
| --- |

Cài đặt matplotlib:

| $ pip install matplotlib  Collecting matplotlib  Using cached matplotlib-3.7.1-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (11.6 MB)  Collecting python-dateutil>=2.7  Using cached python\_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)  Collecting numpy>=1.20  Using cached numpy-1.24.3-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (17.3 MB)  Collecting packaging>=20.0  Using cached packaging-23.1-py3-none-any.whl (48 kB)  Collecting cycler>=0.10  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)  Collecting pyparsing>=2.3.1  Using cached pyparsing-3.0.9-py3-none-any.whl (98 kB)  Collecting fonttools>=4.22.0  Downloading fonttools-4.39.4-py3-none-any.whl (1.0 MB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 1.9 MB/s eta 0:00:00  Collecting contourpy>=1.0.1  Downloading contourpy-1.0.7-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (300 kB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 300.3/300.3 kB 2.0 MB/s eta 0:00:00  Collecting pillow>=6.2.0  Using cached Pillow-9.5.0-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (3.3 MB)  Collecting kiwisolver>=1.0.1  Downloading kiwisolver-1.4.4-cp310-cp310-manylinux\_2\_12\_x86\_64.manylinux2010\_x86\_64.whl (1.6 MB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 2.6 MB/s eta 0:00:00  Collecting six>=1.5  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)  Installing collected packages: six, pyparsing, pillow, packaging, numpy, kiwisolver, fonttools, cycler, python-dateutil, contourpy, matplotlib  Successfully installed contourpy-1.0.7 cycler-0.11.0 fonttools-4.39.4 kiwisolver-1.4.4 matplotlib-3.7.1 numpy-1.24.3 packaging-23.1 pillow-9.5.0 pyparsing-3.0.9 python-dateutil-2.8.2 six-1.16.0 |
| --- |

Cài đặt một số thư viện cần thiết khác:

| $ pip install hyperopt  Collecting hyperopt  Downloading hyperopt-0.2.7-py2.py3-none-any.whl (1.6 MB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 9.2 MB/s eta 0:00:00  Requirement already satisfied: six in /root/anaconda3/envs/contrastive\_learning/lib/python3.10/site-packages (from hyperopt) (1.16.0)  Collecting py4j  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 200.5/200.5 kB 4.8 MB/s eta 0:00:00  Collecting tqdm  Using cached tqdm-4.65.0-py3-none-any.whl (77 kB)  Requirement already satisfied: networkx>=2.2 in /root/anaconda3/envs/contrastive\_learning/lib/python3.10/site-packages (from hyperopt) (3.1)  Requirement already satisfied: scipy in /root/anaconda3/envs/contrastive\_learning/lib/python3.10/site-packages (from hyperopt) (1.10.1)  Requirement already satisfied: numpy in /root/anaconda3/envs/contrastive\_learning/lib/python3.10/site-packages (from hyperopt) (1.24.3)  Collecting future  Downloading future-0.18.3.tar.gz (840 kB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 840.9/840.9 kB 20.6 MB/s eta 0:00:00  Preparing metadata (setup.py) ... done  Collecting cloudpickle  Using cached cloudpickle-2.2.1-py3-none-any.whl (25 kB)  Building wheels for collected packages: future  Building wheel for future (setup.py) ... done  Created wheel for future: filename=future-0.18.3-py3-none-any.whl size=492025 sha256=c9642ac7e2ff678ca228fb58fe5636d26221f65d7d41b7151cf513494c59431b  Stored in directory: /root/.cache/pip/wheels/5e/a9/47/f118e66afd12240e4662752cc22cefae5d97275623aa8ef57d  Successfully built future  Installing collected packages: py4j, tqdm, future, cloudpickle, hyperopt  Successfully installed cloudpickle-2.2.1 future-0.18.3 hyperopt-0.2.7 py4j-0.10.9.7 tqdm-4.65.0  $ pip install scikit\_learn  Collecting scikit\_learn  Using cached scikit\_learn-1.2.2-cp310-cp310-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (9.6 MB)  Collecting joblib>=1.1.1  Using cached joblib-1.2.0-py3-none-any.whl (297 kB)  Collecting threadpoolctl>=2.0.0  Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)  Requirement already satisfied: scipy>=1.3.2 in /root/anaconda3/envs/contrastive\_learning/lib/python3.10/site-packages (from scikit\_learn) (1.10.1)  Requirement already satisfied: numpy>=1.17.3 in /root/anaconda3/envs/contrastive\_learning/lib/python3.10/site-packages (from scikit\_learn) (1.24.3)  Installing collected packages: threadpoolctl, joblib, scikit\_learn  Successfully installed joblib-1.2.0 scikit\_learn-1.2.2 threadpoolctl-3.1.0  $ pip install pyyaml  Collecting pyyaml  Downloading PyYAML-6.0-cp310-cp310-manylinux\_2\_5\_x86\_64.manylinux1\_x86\_64.manylinux\_2\_12\_x86\_64.manylinux2010\_x86\_64.whl (682 kB)  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 682.2/682.2 kB 4.9 MB/s eta 0:00:00  Installing collected packages: pyyaml  Successfully installed pyyaml-6.0  $ pip install colorlog  Collecting colorlog  Downloading colorlog-6.7.0-py2.py3-none-any.whl (11 kB)  Installing collected packages: colorlog  Successfully installed colorlog-6.7.0  $ pip install colorama  Collecting colorama  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)  Installing collected packages: colorama  Successfully installed colorama-0.4.6 |
| --- |

### 3.3.2. Sinh tập dữ liệu

chúng ta có thể sử dụng tập movielens với nhiều tập dữ liệu có số user, item lớn khác nhau như:

* MovieLens 100K Dataset : 100,000 sample, 944 users, 1350 movies
* MovieLens 1M Dataset: 1 million sample, from 6041 users on 3417 movies

### 3.3.3. Xử lý tập dữ liệu

### 

| def data\_preparation(config, dataset, save=False):  *"""Split the dataset by :attr:`config['eval\_setting']` and call :func:`dataloader\_construct` to create*  *corresponding dataloader.*  *Args:*  *config (Config): An instance object of Config, used to record parameter information.*  *dataset (Dataset): An instance object of Dataset, which contains all interaction records.*  *save (bool, optional): If ``True``, it will call :func:`save\_datasets` to save split dataset.*  *Defaults to ``False``.*  *Returns:*  *tuple:*  *- train\_data (AbstractDataLoader): The dataloader for training.*  *- valid\_data (AbstractDataLoader): The dataloader for validation.*  *- test\_data (AbstractDataLoader): The dataloader for testing.*  *"""*  model\_type = config['MODEL\_TYPE']  es = EvalSetting(config)  built\_datasets = dataset.build(es)  train\_dataset, valid\_dataset, test\_dataset = built\_datasets  phases = ['train', 'valid', 'test']  sampler = None  logger = getLogger()  train\_neg\_sample\_args = config['train\_neg\_sample\_args']  eval\_neg\_sample\_args = es.neg\_sample\_args  # Training  train\_kwargs = {  'config': config,  'dataset': train\_dataset,  'batch\_size': config['train\_batch\_size'],  'dl\_format': config['MODEL\_INPUT\_TYPE'],  'shuffle': True,  'phase': 'train'  }  if train\_neg\_sample\_args['strategy'] != 'none':  if dataset.label\_field in dataset.inter\_feat:  raise ValueError(  f'`training\_neg\_sample\_num` should be 0 '  f'if inter\_feat have label\_field [{dataset.label\_field}].'  )  if model\_type != ModelType.SEQUENTIAL:  sampler = Sampler(phases, built\_datasets, train\_neg\_sample\_args['distribution'])  else:  sampler = RepeatableSampler(phases, dataset, train\_neg\_sample\_args['distribution'])  train\_kwargs['sampler'] = sampler.set\_phase('train')  train\_kwargs['neg\_sample\_args'] = train\_neg\_sample\_args  if model\_type == ModelType.KNOWLEDGE:  kg\_sampler = KGSampler(dataset, train\_neg\_sample\_args['distribution'])  train\_kwargs['kg\_sampler'] = kg\_sampler  dataloader = get\_data\_loader('train', config, train\_neg\_sample\_args)  logger.info(  set\_color('Build', 'pink') + set\_color(f' [{dataloader.\_\_name\_\_}]', 'yellow') + ' for ' +  set\_color('[train]', 'yellow') + ' with format ' + set\_color(f'[{train\_kwargs["dl\_format"]}]', 'yellow')  )  if train\_neg\_sample\_args['strategy'] != 'none':  logger.info(  set\_color('[train]', 'pink') + set\_color(' Negative Sampling', 'blue') + f': {train\_neg\_sample\_args}'  )  else:  logger.info(set\_color('[train]', 'pink') + set\_color(' No Negative Sampling', 'yellow'))  logger.info(  set\_color('[train]', 'pink') + set\_color(' batch\_size', 'cyan') + ' = ' +  set\_color(f'[{train\_kwargs["batch\_size"]}]', 'yellow') + ', ' + set\_color('shuffle', 'cyan') + ' = ' +  set\_color(f'[{train\_kwargs["shuffle"]}]\n', 'yellow')  )  train\_data = dataloader(\*\*train\_kwargs)  # Evaluation  eval\_kwargs = {  'config': config,  'batch\_size': config['eval\_batch\_size'],  'dl\_format': InputType.POINTWISE,  'shuffle': False,  'phase': 'eval'  }  valid\_kwargs = {'dataset': valid\_dataset}  test\_kwargs = {'dataset': test\_dataset}  if eval\_neg\_sample\_args['strategy'] != 'none':  if dataset.label\_field in dataset.inter\_feat:  raise ValueError(  f'It can not validate with `{es.es\_str[1]}` '  f'when inter\_feat have label\_field [{dataset.label\_field}].'  )  if sampler is None:  if model\_type != ModelType.SEQUENTIAL:  sampler = Sampler(phases, built\_datasets, eval\_neg\_sample\_args['distribution'])  else:  sampler = RepeatableSampler(phases, dataset, eval\_neg\_sample\_args['distribution'])  else:  sampler.set\_distribution(eval\_neg\_sample\_args['distribution'])  eval\_kwargs['neg\_sample\_args'] = eval\_neg\_sample\_args  valid\_kwargs['sampler'] = sampler.set\_phase('valid')  test\_kwargs['sampler'] = sampler.set\_phase('test')  valid\_kwargs.update(eval\_kwargs)  test\_kwargs.update(eval\_kwargs)  dataloader = get\_data\_loader('evaluation', config, eval\_neg\_sample\_args)  logger.info(  set\_color('Build', 'pink') + set\_color(f' [{dataloader.\_\_name\_\_}]', 'yellow') + ' for ' +  set\_color('[evaluation]', 'yellow') + ' with format ' + set\_color(f'[{eval\_kwargs["dl\_format"]}]', 'yellow')  )  logger.info(es)  logger.info(  set\_color('[evaluation]', 'pink') + set\_color(' batch\_size', 'cyan') + ' = ' +  set\_color(f'[{eval\_kwargs["batch\_size"]}]', 'yellow') + ', ' + set\_color('shuffle', 'cyan') + ' = ' +  set\_color(f'[{eval\_kwargs["shuffle"]}]\n', 'yellow')  )  valid\_data = dataloader(\*\*valid\_kwargs)  test\_data = dataloader(\*\*test\_kwargs)  if save:  save\_split\_dataloaders(config, dataloaders=(train\_data, valid\_data, test\_data))  return train\_data, valid\_data, test\_data |
| --- |

### 3.3.4. Quá trình huấn luyện mô hình

### 

| # model training  best\_valid\_score, best\_valid\_result = trainer.fit(  train\_data, valid\_data, saved=saved, show\_progress=config['show\_progress']  )  import numpy as np  import seaborn as sns  import matplotlib.pyplot as plt  from sklearn.decomposition import TruncatedSVD  embedding\_matrix = model.item\_embedding.weight[1:].cpu().detach().numpy()  svd = TruncatedSVD(n\_components=2)  svd.fit(embedding\_matrix)  comp\_tr = np.transpose(svd.components\_)  proj = np.dot(embedding\_matrix, comp\_tr)  cnt = {}  for i in dataset['item\_id']:  if i.item() in cnt:  cnt[i.item()] += 1  else:  cnt[i.item()] = 1  freq = np.zeros(embedding\_matrix.shape[0])  for i in cnt:  freq[i-1] = cnt[i]  # freq /= freq.max()  sns.set(style='darkgrid')  sns.set\_context("notebook", font\_scale=1.8, rc={"lines.linewidth": 3, 'lines.markersize': 20})  plt.figure(figsize=(6, 4.5))  plt.scatter(proj[:, 0], proj[:, 1], s=1, c=freq, cmap='viridis\_r')  plt.colorbar()  plt.xlim(-2, 2)  plt.ylim(-2, 2)  # plt.axis('square')  # plt.show()  plt.savefig(log\_dir + '/' + config['model'] + '-' + config['dataset'] + '.pdf', format='pdf', transparent=False, bbox\_inches='tight')  from scipy.linalg import svdvals  svs = svdvals(embedding\_matrix)  svs /= svs.max()  np.save(log\_dir + '/sv.npy', svs)  sns.set(style='darkgrid')  sns.set\_context("notebook", font\_scale=1.8, rc={"lines.linewidth": 3, 'lines.markersize': 20})  plt.figure(figsize=(6, 4.5))  plt.plot(svs)  # plt.show()  plt.savefig(log\_dir + '/svs.pdf', format='pdf', transparent=False, bbox\_inches='tight') |
| --- |

| General Hyper Parameters:  gpu\_id = 0  use\_gpu = True  seed = 2020  state = INFO  reproducibility = True  data\_path = /home/viettel\_May23/contrastive\_learning/DuoRec/recbole/config/../dataset\_example/ml-100k  show\_progress = True  Training Hyper Parameters:  checkpoint\_dir = saved  epochs = 50  train\_batch\_size = 512  learner = adam  learning\_rate = 0.001  training\_neg\_sample\_num = 0  training\_neg\_sample\_distribution = uniform  eval\_step = 1  stopping\_step = 10  clip\_grad\_norm = None  weight\_decay = 0  draw\_loss\_pic = False  loss\_decimal\_place = 4  Evaluation Hyper Parameters:  eval\_setting = TO\_LS,full  group\_by\_user = True  split\_ratio = [0.8, 0.1, 0.1]  leave\_one\_num = 2  real\_time\_process = False  metrics = ['Recall', 'MRR', 'NDCG', 'Precision']  topk = [5, 10, 20, 50]  valid\_metric = MRR@10  eval\_batch\_size = 256  metric\_decimal\_place = 4  Dataset Hyper Parameters:  field\_separator =  seq\_separator =  USER\_ID\_FIELD = user\_id  ITEM\_ID\_FIELD = item\_id  RATING\_FIELD = rating  TIME\_FIELD = timestamp  seq\_len = None  LABEL\_FIELD = label  threshold = None  NEG\_PREFIX = neg\_  load\_col = {'inter': ['user\_id', 'item\_id', 'rating', 'timestamp']}  unload\_col = None  unused\_col = None  additional\_feat\_suffix = None  rm\_dup\_inter = None  lowest\_val = None  highest\_val = None  equal\_val = None  not\_equal\_val = None  filter\_inter\_by\_user\_or\_item = True  max\_user\_inter\_num = None  min\_user\_inter\_num = 5  max\_item\_inter\_num = None  min\_item\_inter\_num = 5  fields\_in\_same\_space = None  preload\_weight = None  normalize\_field = None  normalize\_all = True  ITEM\_LIST\_LENGTH\_FIELD = item\_length  LIST\_SUFFIX = \_list  MAX\_ITEM\_LIST\_LENGTH = 50  POSITION\_FIELD = position\_id  HEAD\_ENTITY\_ID\_FIELD = head\_id  TAIL\_ENTITY\_ID\_FIELD = tail\_id  RELATION\_ID\_FIELD = relation\_id  ENTITY\_ID\_FIELD = entity\_id  Other Hyper Parameters:  valid\_metric\_bigger = True  n\_layers = 2  n\_heads = 2  hidden\_size = 64  inner\_size = 256  hidden\_dropout\_prob = 0.5  attn\_dropout\_prob = 0.5  hidden\_act = gelu  layer\_norm\_eps = 1e-12  initializer\_range = 0.02  loss\_type = CE  lmd = 0.1  lmd\_sem = 0.1  SSL\_AUG = DuoRec  SOURCE\_ID\_FIELD = source\_id  TARGET\_ID\_FIELD = target\_id  benchmark\_filename = None  MODEL\_TYPE = ModelType.SEQUENTIAL  log\_root = ./log/  tau = 1  contrast = us\_x  sim = dot  MODEL\_INPUT\_TYPE = InputType.POINTWISE  eval\_type = EvaluatorType.RANKING  device = cpu  train\_neg\_sample\_args = {'strategy': 'none'}  log\_dir = /home/viettel\_May23/contrastive\_learning/DuoRec/log/DuoRec/ml-100k/bs512-lmd0.1-sem0.1-us\_x-May-22-2023\_04-47-37-lr0.001-l20-tau1-dot-DPh0.5-DPa0.5  ers: 944  Average actions of users: 105.28844114528101  The number of items: 1350  Average actions of items: 73.6004447739066  The number of inters: 99287  The sparsity of the dataset: 92.20911801632141%  Remain Fields: ['user\_id', 'item\_id', 'rating', 'timestamp']  22 May 04:47 INFO Build [SequentialDataLoader] for [train] with format [InputType.POINTWISE]  22 May 04:47 INFO [train] No Negative Sampling  22 May 04:47 INFO [train] batch\_size = [512], shuffle = [True]  22 May 04:47 INFO Build [SequentialFullDataLoader] for [evaluation] with format [InputType.POINTWISE]  22 May 04:47 INFO Evaluation Setting:  Group by user\_id  Ordering: {'strategy': 'by', 'field': 'timestamp', 'ascending': True}  Splitting: {'strategy': 'loo', 'leave\_one\_num': 2}  Negative Sampling: {'strategy': 'full', 'distribution': 'uniform'}  22 May 04:47 INFO [evaluation] batch\_size = [256], shuffle = [False]  22 May 04:47 INFO DuoRec(  (item\_embedding): Embedding(1350, 64, padding\_idx=0)  (position\_embedding): Embedding(50, 64)  (trm\_encoder): TransformerEncoder(  (layer): ModuleList(  (0-1): 2 x TransformerLayer(  (multi\_head\_attention): MultiHeadAttention(  (query): Linear(in\_features=64, out\_features=64, bias=True)  (key): Linear(in\_features=64, out\_features=64, bias=True)  (value): Linear(in\_features=64, out\_features=64, bias=True)  (attn\_dropout): Dropout(p=0.5, inplace=False)  (dense): Linear(in\_features=64, out\_features=64, bias=True)  (LayerNorm): LayerNorm((64,), eps=1e-12, elementwise\_affine=True)  (out\_dropout): Dropout(p=0.5, inplace=False)  )  (feed\_forward): FeedForward(  (dense\_1): Linear(in\_features=64, out\_features=256, bias=True)  (dense\_2): Linear(in\_features=256, out\_features=64, bias=True)  (LayerNorm): LayerNorm((64,), eps=1e-12, elementwise\_affine=True)  (dropout): Dropout(p=0.5, inplace=False)  )  )  )  )  (LayerNorm): LayerNorm((64,), eps=1e-12, elementwise\_affine=True)  (dropout): Dropout(p=0.5, inplace=False)  (loss\_fct): CrossEntropyLoss()  (aug\_nce\_fct): CrossEntropyLoss()  (sem\_aug\_nce\_fct): CrossEntropyLoss()  )  Trainable parameters: 189696 |
| --- |



## 3.4 Build mô hình LSTM

### 3.4.1. Sinh tập dữ liệu

| def main(args):  # Get parse  num\_feature\_extend = args.num\_f  num\_row = args.num\_row  np.random.seed(42)  # read sample data  sample\_data = pd.read\_csv("./data/sp50k.csv")  print(sample\_data.shape)  if num\_row > 0:  print('gen new sample')  new\_data = np.random.rand(num\_row, sample\_data.shape[1])  print(new\_data.shape)  print(sample\_data.shape)  new\_data = pd.DataFrame(new\_data, columns=sample\_data.columns)  df\_merged = sample\_data.append(new\_data, ignore\_index=True)  print(df\_merged.shape)  df\_merged.to\_csv("./data/sp50k.csv")  # print(sample\_data.head(5))  # gen num feature  # num\_feature\_extend = 5  # list\_feature\_extend = [f"field{i+3}" for i in range(num\_feature\_extend)]  # print(list\_feature\_extend)  for i in range(num\_feature\_extend):  new\_col = "field" + str(i + 3)  sample\_data[new\_col] = np.random.randint(1, 10 \*\* 9, sample\_data.shape[0])  total\_f = num\_feature\_extend + 2  print(sample\_data.head(3))  # Save new data  sample\_data.to\_csv(f"./data/sp50k\_{total\_f}feature.csv")  if \_\_name\_\_ == "\_\_main\_\_":  parser = argparse.ArgumentParser(description="Generator data for LSTM model")  parser.add\_argument("num\_f", type=int, default=0, help="num of new feature")  parser.add\_argument("num\_row", type=int, default=0, help="num of new feature")  main(parser.parse\_args()) |
| --- |

### 3.4.2. Xử lý tập dữ liệu

| class DataLoader():  *"""A class for loading and transforming data for the lstm model"""*  def \_\_init\_\_(self, filename, split, cols):  dataframe = pd.read\_csv(filename)  i\_split = int(len(dataframe) \* split)  self.data\_train = dataframe.get(cols).values[:i\_split]  self.data\_test = dataframe.get(cols).values[i\_split:]  self.len\_train = len(self.data\_train)  self.len\_test = len(self.data\_test)  self.len\_train\_windows = None  def get\_test\_data(self, seq\_len, normalise):  *'''*  *Create x, y test data windows*  *Warning: batch method, not generative, make sure you have enough memory to*  *load data, otherwise reduce size of the training split.*  *'''*  data\_windows = []  for i in range(self.len\_test - seq\_len):  data\_windows.append(self.data\_test[i:i + seq\_len])  data\_windows = np.array(data\_windows).astype(float)  data\_windows = self.normalise\_windows(data\_windows, single\_window=False) if normalise else data\_windows  x = data\_windows[:, :-1]  y = data\_windows[:, -1, [0]]  return x, y  def get\_train\_data(self, seq\_len, normalise):  *'''*  *Create x, y train data windows*  *Warning: batch method, not generative, make sure you have enough memory to*  *load data, otherwise use generate\_training\_window() method.*  *'''*  data\_x = []  data\_y = []  for i in range(self.len\_train - seq\_len):  x, y = self.\_next\_window(i, seq\_len, normalise)  data\_x.append(x)  data\_y.append(y)  return np.array(data\_x), np.array(data\_y)  def generate\_train\_batch(self, seq\_len, batch\_size, normalise):  *'''Yield a generator of training data from filename on given list of cols split for train/test'''*  i = 0  while i < (self.len\_train - seq\_len):  x\_batch = []  y\_batch = []  for b in range(batch\_size):  if i >= (self.len\_train - seq\_len):  # stop-condition for a smaller final batch if data doesn't divide evenly  yield np.array(x\_batch), np.array(y\_batch)  i = 0  x, y = self.\_next\_window(i, seq\_len, normalise)  x\_batch.append(x)  y\_batch.append(y)  i += 1  yield np.array(x\_batch), np.array(y\_batch)  def \_next\_window(self, i, seq\_len, normalise):  *'''Generates the next data window from the given index location i'''*  window = self.data\_train[i:i + seq\_len]  window = self.normalise\_windows(window, single\_window=True)[0] if normalise else window  x = window[:-1]  y = window[-1, [0]]  return x, y  def normalise\_windows(self, window\_data, single\_window=False):  *'''Normalise window with a base value of zero'''*  normalised\_data = []  window\_data = [window\_data] if single\_window else window\_data  for window in window\_data:  normalised\_window = []  for col\_i in range(window.shape[1]):  normalised\_col = [((float(p) / float(window[0, col\_i])) - 1) for p in window[:, col\_i]]  normalised\_window.append(normalised\_col)  normalised\_window = np.array(  normalised\_window).T # reshape and transpose array back into original multidimensional format  normalised\_data.append(normalised\_window)  return np.array(normalised\_data) |
| --- |

### 3.4.3. Quá trình huấn luyện mô hình

| # python run.py  Using TensorFlow backend.  [Model] Model Compiled  Time taken: 0:00:00.627034  [Model] Training Started  [Model] 2 epochs, 32 batch size, 124 batches per epoch  Epoch 1/2  124/124 [==============================] - 20s 158ms/step - loss: 0.0018  Epoch 2/2  124/124 [==============================] - 19s 152ms/step - loss: 6.7709e-04  [Model] Training Completed. Model saved as saved\_models/21052023-190434-e2.h5  Time taken: 0:00:40.653776  [Model] Predicting Sequences Multiple... |
| --- |

# IV. Fine-Tune và đánh giá performance nhóm mô hình Machine learning

Trong lân thử nghiệm này, tôi tập trung vào đánh giá hiệu suất của thiết bị phần cứng trong quá trình huấn luyện và đánh giá mô hình, chưa tập trung đến độ chính xác cung cấp bởi mô hình.

## 4.1. Fine-Tune và đánh giá performance nhóm mô hình Linear Regression

### 4.1.1. Test mô hình

### 

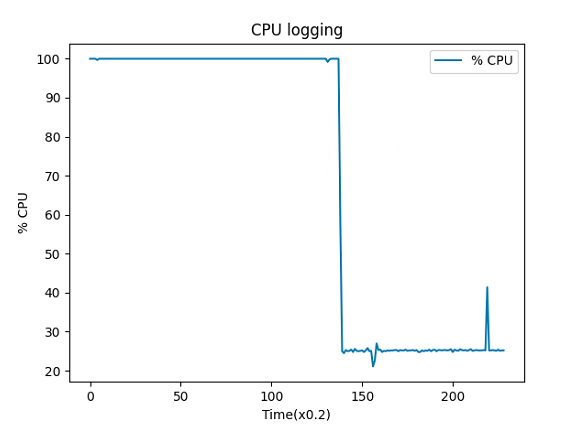
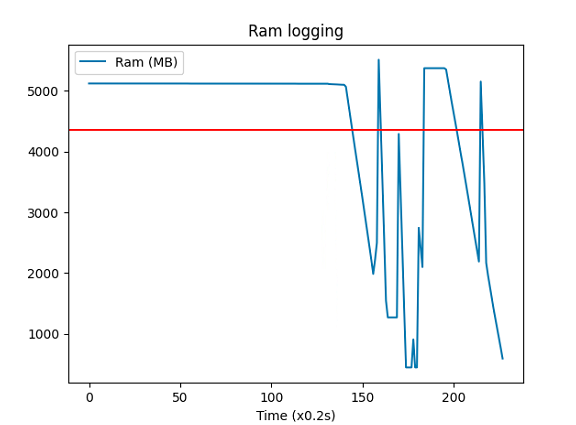
| # Fitting  def fitting(X\_train, y\_train):  lm = LinearRegression(n\_jobs=1)  lm.fit(X\_train, y\_train)  del X\_train  del y\_train  return lm  # Saving model  def save(lm):  with open('LinearModel.sav', mode='wb') as f:  pickle.dump(lm, f)  # Inference  def model\_run(model, testfile):  *"""*  *Loads and runs a sklearn model*  *"""*  lm = pickle.load(open(model, 'rb'))  X\_test = pd.read\_csv(testfile)  \_ = lm.predict(X\_test)  return None  NUM\_SAMPLES, NUM\_FEATURES = 20000, 10000  data = make\_data(NUM\_SAMPLES, NUM\_FEATURES)  # X\_train, y\_train = data.iloc[0:].drop(['output'], axis=1), data.iloc[0:]['output']  X\_train, y\_train, X\_test, y\_test = test\_train(data, NUM\_SAMPLES)  X\_test.to\_csv("Test.csv", index=False)  print("Training model")  lm = fitting(X\_train, y\_train)  print("Saving and Inference model")  save(lm)  model\_run('LinearModel.sav', 'Test.csv') |
| --- |

### 4.1.2.Đánh giá performance

#### 4.3.2.1 Đánh giá performance lần 1

Data info:

| NUM\_SAMPLES | 20000 |
| --- | --- |
| NUM\_FEATURES | 20000 |



## 4.2. Fine-Tune và đánh giá performance nhóm mô hình Logistic Regression

### 4.2.1. Test mô hình

### 

| # Fitting  def fitting(X\_train, y\_train):  lm = LogisticRegression(n\_jobs=1)  lm.fit(X\_train, y\_train)  del X\_train  del y\_train  return lm  # Saving model  def save(lm):  with open('LogisticModel.sav', mode='wb') as f:  pickle.dump(lm, f)  # Inference  def model\_run(model, testfile):  *"""*  *Loads and runs a sklearn model*  *"""*  lm = pickle.load(open(model, 'rb'))  X\_test = pd.read\_csv(testfile)  \_ = lm.predict(X\_test)  return None  NUM\_SAMPLES, NUM\_FEATURES = 20000, 10000  data = make\_data(NUM\_SAMPLES, NUM\_FEATURES)  # X\_train, y\_train = data.iloc[0:].drop(['output'], axis=1), data.iloc[0:]['output']  X\_train, y\_train, X\_test, y\_test = test\_train(data, NUM\_SAMPLES)  X\_test.to\_csv("Test.csv", index=False)  print("Training model")  lm = fitting(X\_train, y\_train)  print("Saving and Inference model")  save(lm)  model\_run('LogisticModel.sav', 'Test.csv') |
| --- |

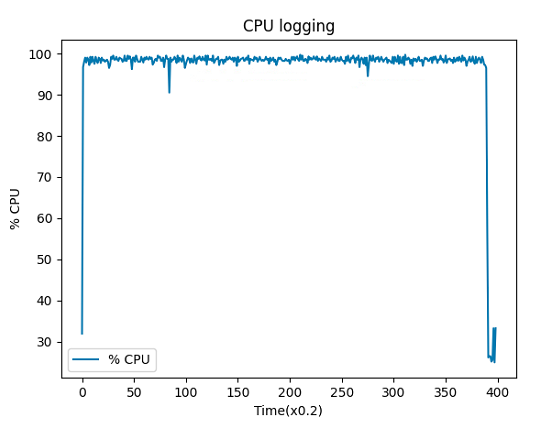
### 4.2.2.Đánh giá performance

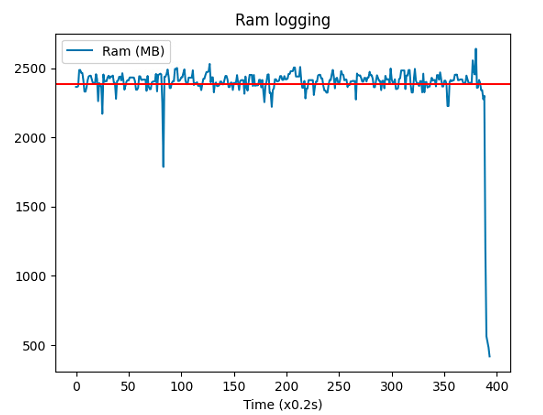
#### 4.3.2.1 Đánh giá performance lần 1

Data size lớn hơn như sau có thể làm Out of Memory bộ nhớ

Data info:

| NUM\_SAMPLES | 10000 |
| --- | --- |
| NUM\_FEATURES | 10000 |





## 4.3. Fine-Tune và đánh giá performance nhóm mô hình SVM

### 4.3.1. Test mô hình

### 

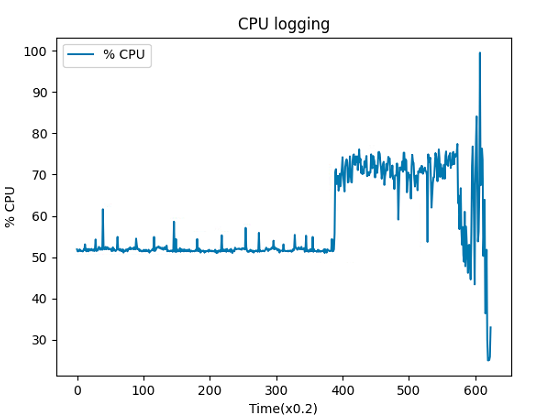
| # Fitting  def fitting(X\_train, y\_train):  lm = SVR(C=1.0, epsilon=0.2)  lm.fit(X\_train, y\_train)  del X\_train  del y\_train  return lm  # Saving model  def save(lm):  with open('SvrModel.sav', mode='wb') as f:  pickle.dump(lm, f)  # Inference  def model\_run(model, testfile):  *"""*  *Loads and runs a sklearn model*  *"""*  lm = pickle.load(open(model, 'rb'))  X\_test = pd.read\_csv(testfile)  \_ = lm.predict(X\_test)  return None  NUM\_SAMPLES, NUM\_FEATURES = 20000, 10000  data = make\_data(NUM\_SAMPLES, NUM\_FEATURES)  # X\_train, y\_train = data.iloc[0:].drop(['output'], axis=1), data.iloc[0:]['output']  X\_train, y\_train, X\_test, y\_test = test\_train(data, NUM\_SAMPLES)  X\_test.to\_csv("Test.csv", index=False)  print("Training model")  lm = fitting(X\_train, y\_train)  print("Saving and Inference model")  save(lm)  model\_run('SvrModel.sav', 'Test.csv') |
| --- |

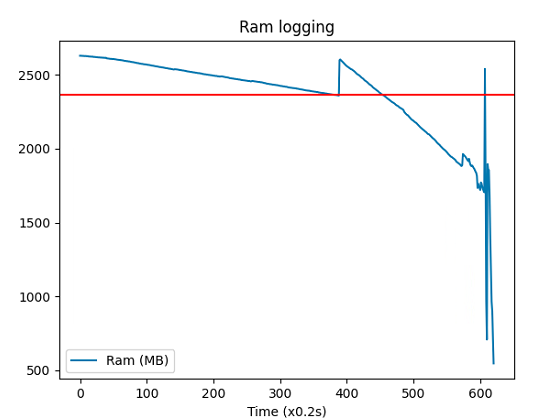
### 4.3.2.Đánh giá performance

#### 4.3.2.1 Đánh giá performance lần 1

Data info:

| NUM\_SAMPLES | 10000 |
| --- | --- |
| NUM\_FEATURES | 10000 |

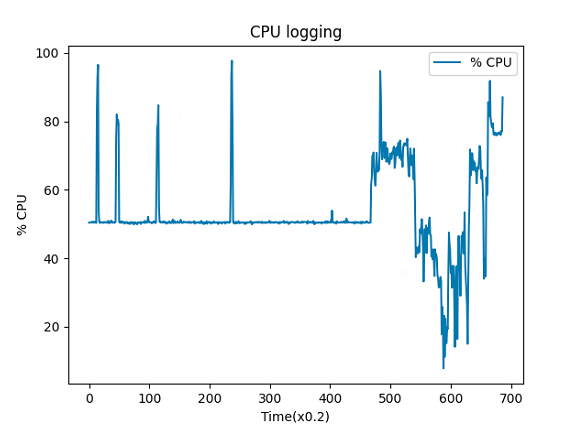


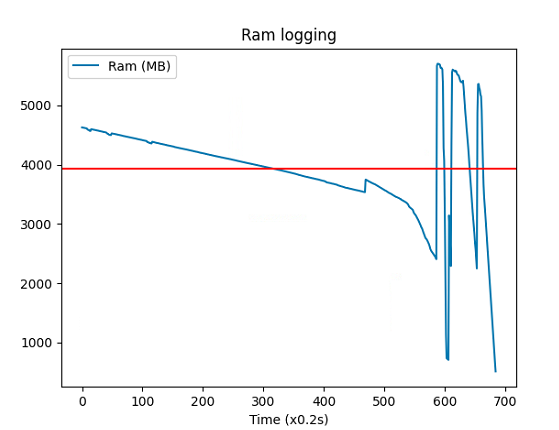


#### 4.3.2.2 Đánh giá performance lần 2

Data info:

| NUM\_SAMPLES | 20000 |
| --- | --- |
| NUM\_FEATURES | 20000 |





# V. Fine-Tune và đánh giá performance nhóm mô hình Deep learning

## 5.1 Fine-Tune và đánh giá performance nhóm mô hình RecVAE

### 5.1.1 Test Mô hình

Trong lân thử nghiệm này, tôi tập trung vào đánh giá hiệu suất của thiết bị phần cứng trong quá trình huấn luyện và đánh giá mô hình, chưa tập trung đến đô chính xác cung cấp bới mô hình.

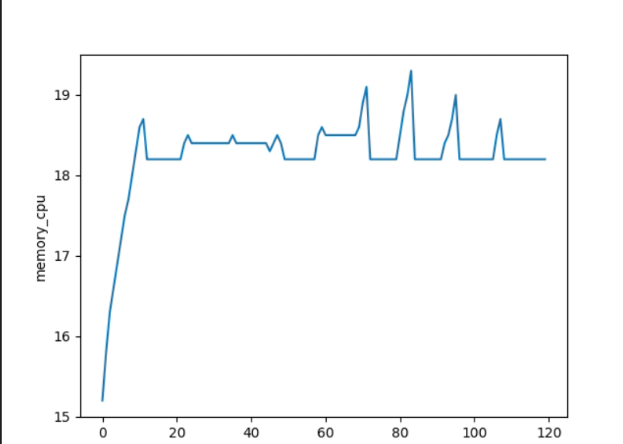
* Tập dữ liệu movieslen-1m:
* =================================================================================
* || NAME || Prec@20 || Prec@50 || Recall@20 || Recall@50 || NDCG@20 || NDCG@50 ||
* ||-----------------------------------------------------------------------------||
* || Score || 0.0129 || 0.0122 || 0.0517 || 0.1220 || 0.0304 || 0.0538 ||
* =================================================================================
* Tập dữ liệu movieslen-10m:
* =================================================================================
* || NAME || Prec@20 || Prec@50 || Recall@20 || Recall@50 || NDCG@20 || NDCG@50 ||
* ||-----------------------------------------------------------------------------||
* || Score || 0.0120 || 0.0161 || 0.0480 || 0.1614 || 0.0268 || 0.0642 ||
* =================================================================================

### 5.1.2. Đánh giá performance

Để đánh giá hiệu suất của CPU khi huấn luyện mô hình, chúng tôi quan tâm đến 2 tham số chính là thời gian huấn luyện và phần trăm ram trong quá trình huấn luyện.

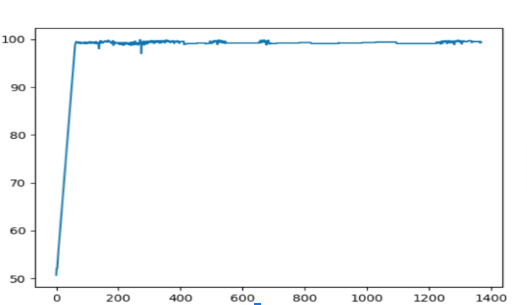
Tổng cộng dung lượng RAM server test: 7,64GB

* Hiệu suất đánh giá của thiết bị trên bộ **movieslen-1m**:
* Thời gian huấn luyện: 28,1s
* Bộ nhớ sử dụng:



**Hiệu suất phần trăm bộ nhớ RAM sử dụng trên movieslen-1m**

Bộ nhớ sử trung trung bình trên thiết bị khoảng 18% ~ 1375Mb

* Hiệu suất đánh giá của thiết bị trên bộ **movieslen-1m:**
* Thời gian huấn luyện: 31 phút 28 giây (=1888.2 sec)
* Bộ nhớ sử dụng:
* 

**Hiệu suất phần trăm bộ nhớ RAM sử dụng trên movieslen-10m**

Trung bình bộ nhớ sử dụng khá lớn chiếm gần hết bộ nhớ, 96.5% bộ nhớ.

## 5.2. Fine-Tune và đánh giá performance nhóm mô hình BERT4REC

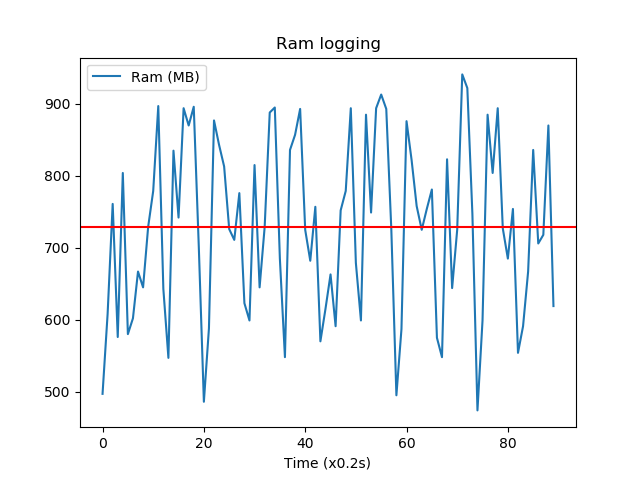
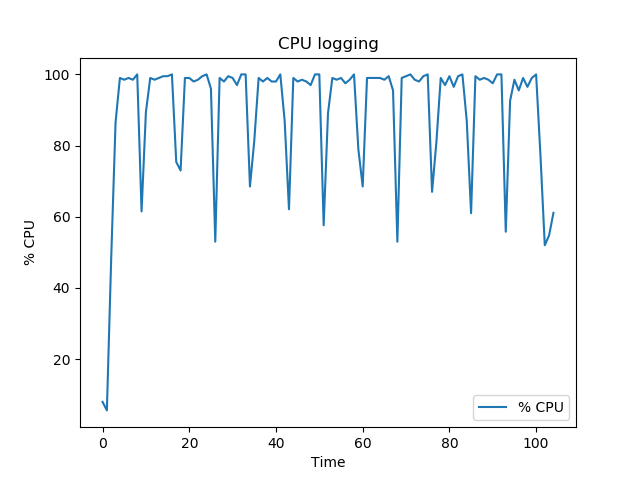
### 5.2.1 Test Mô hình

| if FLAGS.do\_eval:  tf.logging.info("\*\*\*\*\* Running evaluation \*\*\*\*\*")  tf.logging.info(" Batch size = %d", FLAGS.batch\_size)  eval\_input\_fn = input\_fn\_builder(  input\_files=test\_input\_files,  max\_seq\_length=FLAGS.max\_seq\_length,  max\_predictions\_per\_seq=FLAGS.max\_predictions\_per\_seq,  is\_training=False)  #tf.logging.info('special eval ops:', special\_eval\_ops)  result = estimator.evaluate(  input\_fn=eval\_input\_fn,  steps=None,  hooks=[EvalHooks()])  output\_eval\_file = os.path.join(FLAGS.checkpointDir,  "eval\_results.txt")  with tf.gfile.GFile(output\_eval\_file, "w") as writer:  tf.logging.info("\*\*\*\*\* Eval results \*\*\*\*\*")  tf.logging.info(bert\_config.to\_json\_string())  writer.write(bert\_config.to\_json\_string()+'\n')  for key in sorted(result.keys()):  tf.logging.info(" %s = %s", key, str(result[key]))  writer.write("%s = %s\n" % (key, str(result[key]))) |
| --- |

### 5.2.2. Đánh giá performance

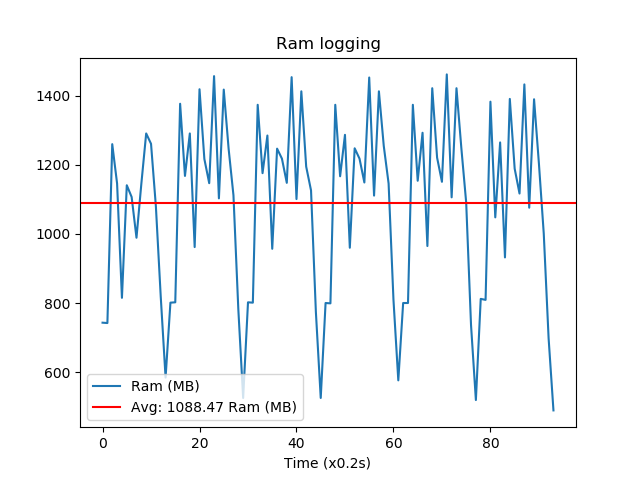
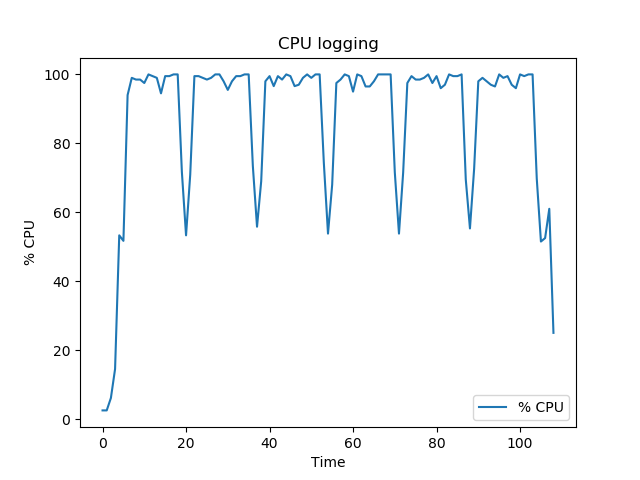
#### 5.2.2.1 Đánh giá performance lần 1

| Step | 10 |
| --- | --- |
| Batch size | 512 |



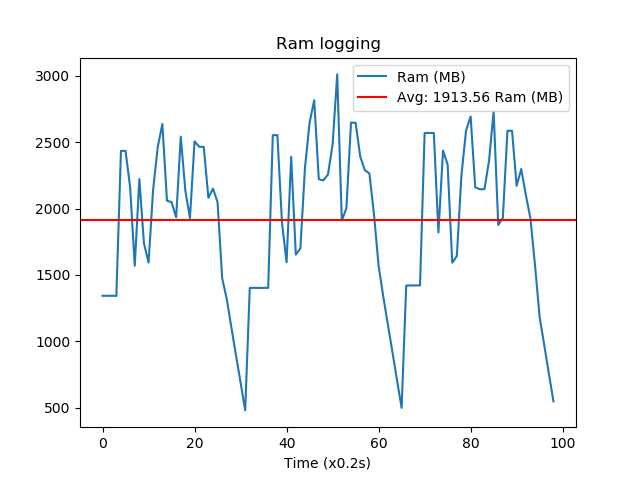
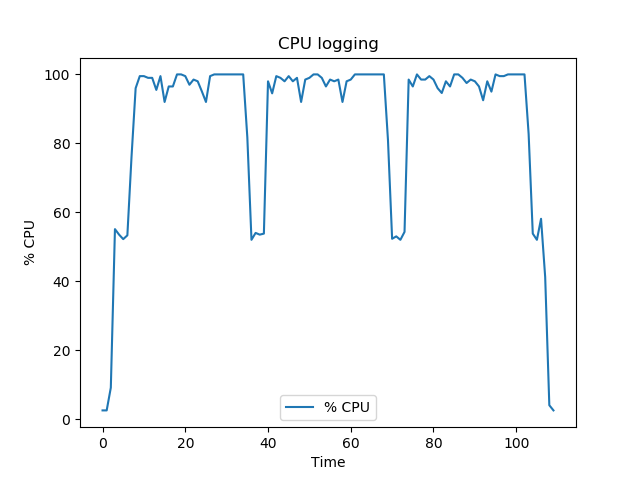
#### 5.2.2.2. Đánh giá performance lần 2

| Step | 10 |
| --- | --- |
| Batch size | 1024 |



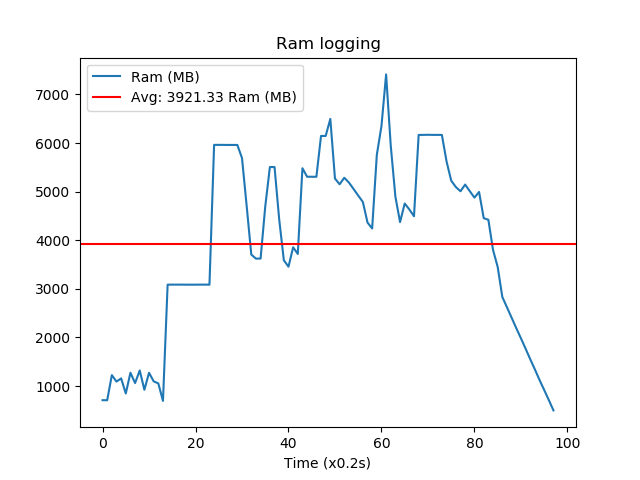
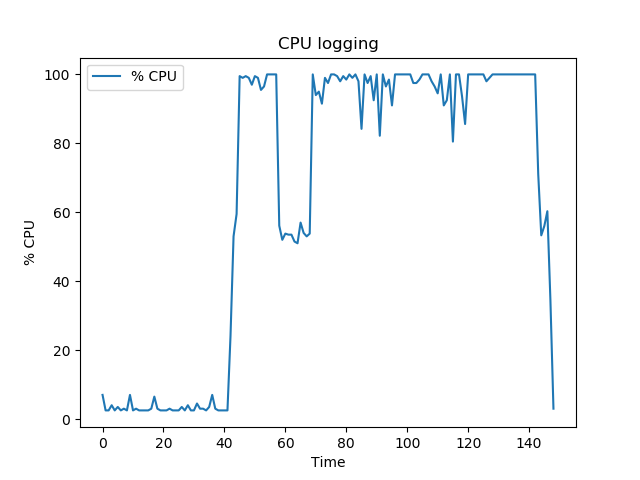
#### 5.2.2.3. Đánh giá performance lần 3

| Step | 10 |
| --- | --- |
| Batch size | 2048 |



#### 5.2.2.4. Đánh giá performance lần 4

| Step | 10 |
| --- | --- |
| Batch size | 5120 |



#### 5.2.2.5. Đánh giá performance lần 5

| Step | 10 |
| --- | --- |
| Batch size | 1024 |

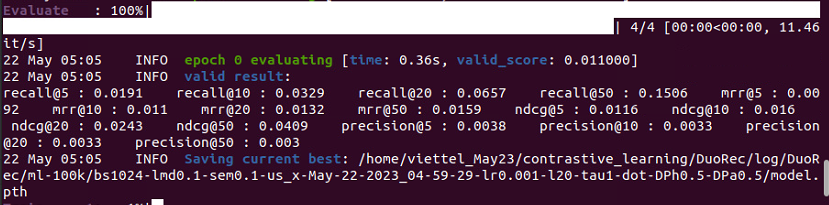
Test trên tập Movielens 20M bị Out of Memory Ram

## 5.3. Fine-Tune và đánh giá performance nhóm mô hình Contrastive Learning

### 5.3.1 Test Mô hình

| # model evaluation  test\_result = trainer.evaluate(test\_data, load\_best\_model=saved, show\_progress=config['show\_progress'])  logger.info(set\_color('best valid ', 'yellow') + f': {best\_valid\_result}')  logger.info(set\_color('test result', 'yellow') + f': {test\_result}') |
| --- |

| Evaluation Hyper Parameters:  eval\_setting = TO\_LS,full  group\_by\_user = True  split\_ratio = [0.8, 0.1, 0.1]  leave\_one\_num = 2  real\_time\_process = False  metrics = ['Recall', 'MRR', 'NDCG', 'Precision']  topk = [5, 10, 20, 50]  valid\_metric = MRR@10  eval\_batch\_size = 256  metric\_decimal\_place = 4  Dataset Hyper Parameters:  field\_separator =  seq\_separator =  USER\_ID\_FIELD = user\_id  ITEM\_ID\_FIELD = item\_id  RATING\_FIELD = rating  TIME\_FIELD = timestamp  seq\_len = None  LABEL\_FIELD = label  threshold = None  NEG\_PREFIX = neg\_  load\_col = {'inter': ['user\_id', 'item\_id', 'rating', 'timestamp']}  unload\_col = None  unused\_col = None  additional\_feat\_suffix = None  lowest\_val = None  highest\_val = None  equal\_val = None  not\_equal\_val = None  max\_user\_inter\_num = None  min\_user\_inter\_num = 5  max\_item\_inter\_num = None  min\_item\_inter\_num = 5  fields\_in\_same\_space = None  preload\_weight = None  normalize\_field = None  normalize\_all = None  ITEM\_LIST\_LENGTH\_FIELD = item\_length  LIST\_SUFFIX = \_list  MAX\_ITEM\_LIST\_LENGTH = 50  POSITION\_FIELD = position\_id  HEAD\_ENTITY\_ID\_FIELD = head\_id  TAIL\_ENTITY\_ID\_FIELD = tail\_id  RELATION\_ID\_FIELD = relation\_id  ENTITY\_ID\_FIELD = entity\_id |
| --- |



### 5.3.2. Đánh giá performance

Để đánh giá hiệu suất của CPU khi huấn luyện mô hình, chúng tôi quan tâm đến 2 tham số chính là thời gian huấn luyện và phần trăm ram trong quá trình huấn luyện.

Tổng cộng dung lượng RAM server test: 7,64GB

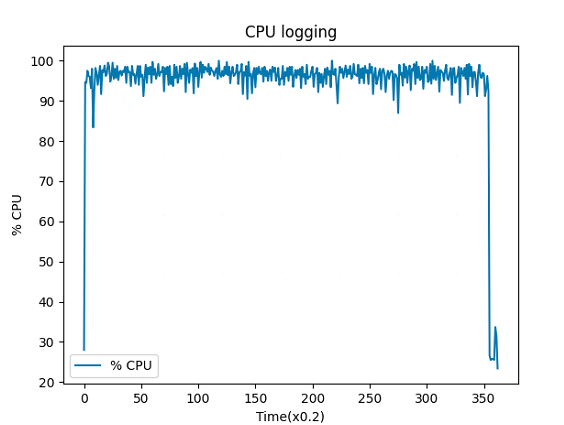
#### 5.3.2.1 Đánh giá performance lần 1

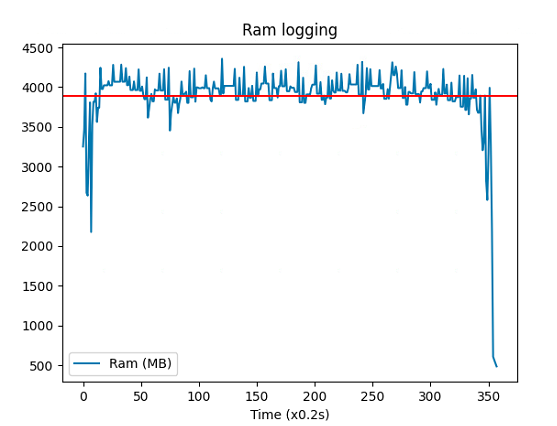
Thông số chi tiết bộ dữ liệu sử dụng để đánh giá:

| INFO ml-100k  The number of users: 944  Average actions of users: 105.28844114528101  The number of items: 1350  Average actions of items: 73.6004447739066  The number of inters: 99287  The sparsity of the dataset: 92.20911801632141%  Remain Fields: ['user\_id', 'item\_id', 'rating', 'timestamp'] |
| --- |

Config:

| batch\_size | 1024 |
| --- | --- |
| epoch | 1 |





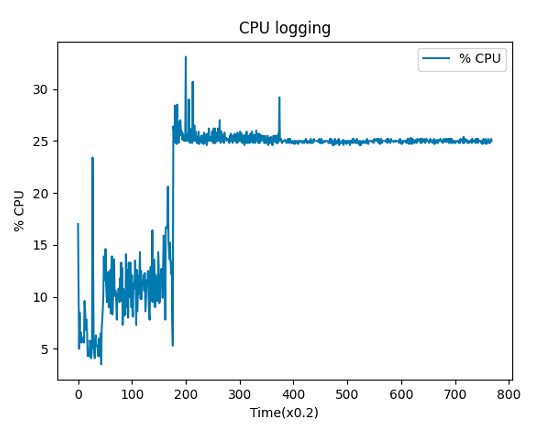
#### 5.3.2.2 Đánh giá performance lần 2

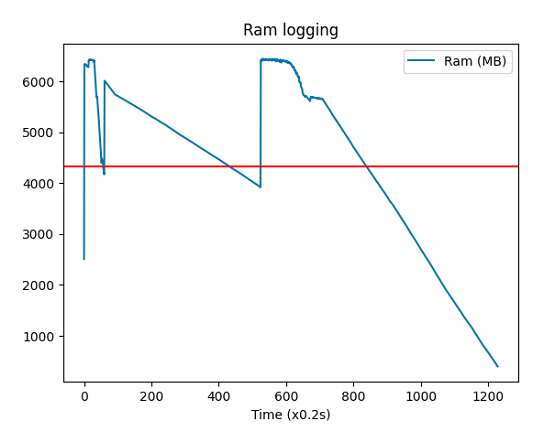
Thông số chi tiết bộ dữ liệu sử dụng để đánh giá:

| INFO ml-1m  The number of users: 6041  Average actions of users: 165.49850993377484  The number of items: 3417  Average actions of items: 292.6261709601874  The number of inters: 999611  The sparsity of the dataset: 95.15741545057172%  Remain Fields: ['user\_id', 'item\_id', 'rating', 'timestamp'] |
| --- |

Config:

| batch\_size | 1024 |
| --- | --- |
| epoch | 1 |

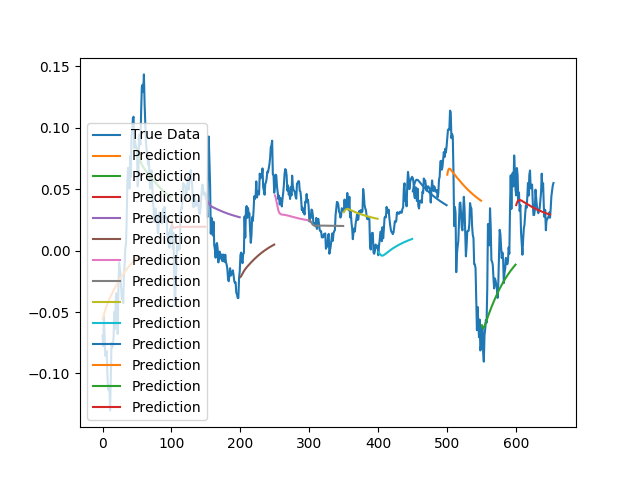




## 5.4. Fine-Tune và đánh giá performance nhóm mô hình LSTM

### 5.4.1 Test Mô hình

| x\_test, y\_test = data.get\_test\_data(  seq\_len=configs['data']['sequence\_length'],  normalise=configs['data']['normalise']  )  predictions = model.predict\_sequences\_multiple(x\_test, configs['data']['sequence\_length'], configs['data']['sequence\_length'])  # predictions = model.predict\_sequence\_full(x\_test, configs['data']['sequence\_length'])  # predictions = model.predict\_point\_by\_point(x\_test)  plot\_results\_multiple(predictions, y\_test, configs['data']['sequence\_length'])  def plot\_results\_multiple(predicted\_data, true\_data, prediction\_len):  fig = plt.figure(facecolor='white')  ax = fig.add\_subplot(111)  ax.plot(true\_data, label='True Data')  # Pad the list of predictions to shift it in the graph to it's correct start  for i, data in enumerate(predicted\_data):  padding = [None for p in range(i \* prediction\_len)]  plt.plot(padding + data, label='Prediction')  plt.legend()  plt.show()  plt.savefig("result.png") |
| --- |



### 5.4.2. Đánh giá performance

#### 5.4.2.1. Đánh giá performance lần 1

Config: config.json

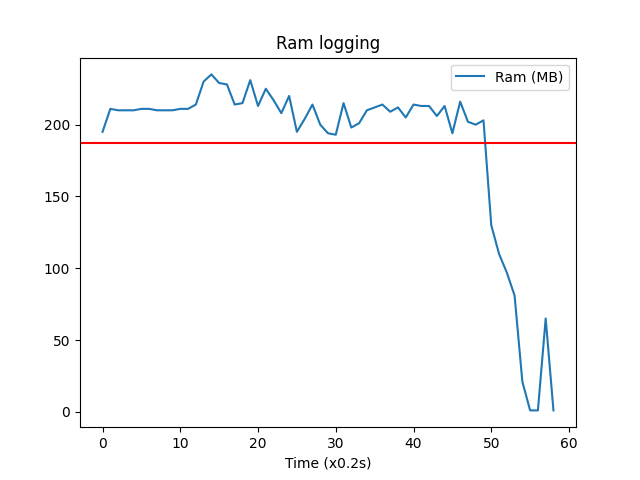
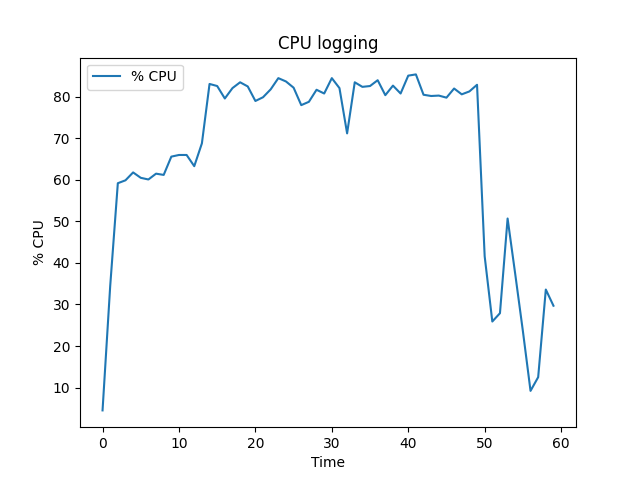
| Số feature | 2 |
| --- | --- |
| batchsize | 32 |
| epochs | 2 |
| Số sample | 5000 |

**Kết quả:**

Time Training: 40.653776s

Dung lượng RAM sử dụng trung bình: 187.22MB

Dung lượng RAM sử dụng cao nhất: 235MB



#### 5.4.2.2. Đánh giá performance lần 2

Config:

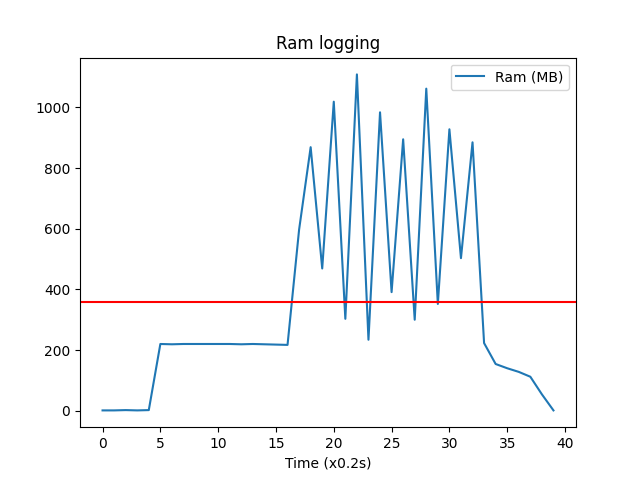
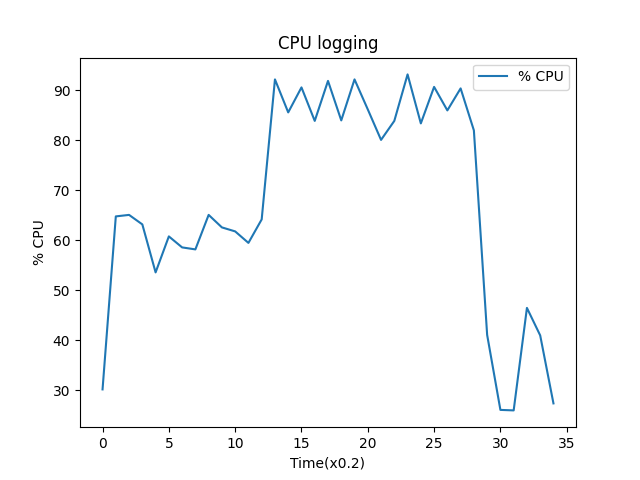
| Số feature | 2 |
| --- | --- |
| batchsize | 1024 |
| Số sample | 5000 |
| Epochs | 2 |

**Kết quả:**

Time Training :19.721629s

Dung lượng RAM sử dụng trung bình: 358.775 MB

Dung lượng RAM sử dụng cao nhất:1109 MB



#### 5.4.2.3. Đánh giá performance lần 3

Config:

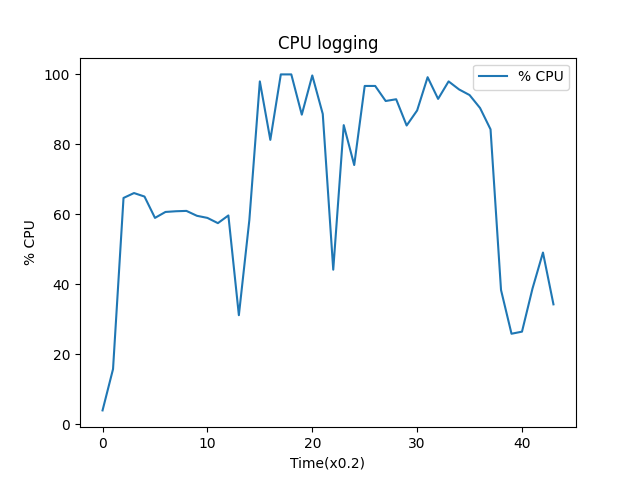
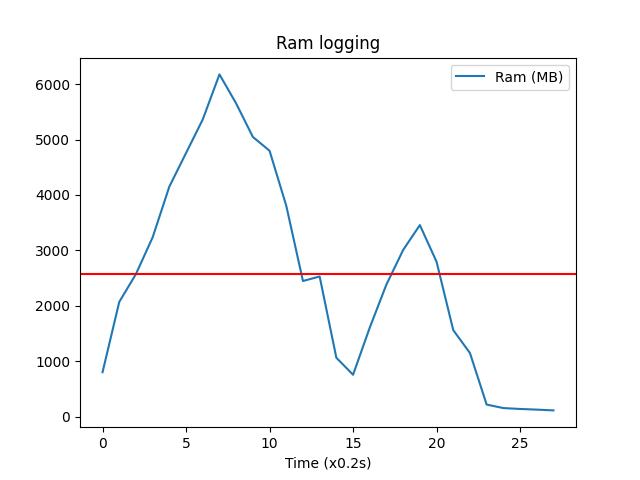
| Số feature | 2 |
| --- | --- |
| batchsize | 7168 |
| Số sample | 50000 |
| Epochs | 2 |

**Kết quả:**

Time Training :162.4168 s

Dung lượng RAM sử dụng trung bình: 3566.769 MB

Dung lượng RAM sử dụng cao nhất: 6405 MB



#### 5.4.2.4. Đánh giá performance lần 4

Config:

| Số feature | 10 |
| --- | --- |
| batchsize | 7168 |
| Số sample | 50000 |
| Epochs | 2 |

**Kết quả:**

Time Training :182.763 s

Dung lượng RAM sử dụng trung bình: 2969.549 MB

Dung lượng RAM sử dụng cao nhất: 7047 MB

