Open Access Full Text Article

Research Article

Long Short-Term Memory Based Movie Recommendation

Duy Bao Tran, Thi Thanh Sang Nguyen*

ABSTRACT

Recommender systems (RS) have become a fundamental tool for helping users make decisions around millions of different choices nowadays – the era of Big Data. It brings a huge benefit for many business models around the world due to their effectiveness on the target customers. A lot of recommendation models and techniques have been proposed and many accomplished incredible outcomes. Collaborative filtering and content-based filtering methods are common, but these both have some disadvantages. A critical one is that they only focus on a user's long-term static preference while ignoring his or her short-term transactional patterns, which results in missing the user's preference shift through the time. In this case, the user's intent at a certain time point may be easily submerged by his or her historical decision behaviors, which leads to unreliable recommendations. To deal with this issue, a session of user interactions with the items can be considered as a solution. In this study, Long Short-Term Memory (LSTM) networks will be analyzed to be applied to user sessions in a recommender system. The MovieLens dataset is considered as a case study of movie recommender systems. This dataset is preprocessed to extract user-movie sessions for user behavior discovery and making movie recommendations to users. Several experiments have been carried out to evaluate the LSTM-based movie recommender system. In the experiments, the LSTM networks are compared with a similar deep learning method, which is Recurrent Neural Networks (RNN), and a baseline machine learning method, which is the collaborative filtering using item-based nearest neighbors (item-KNN). It has been found that the LSTM networks are able to be improved by optimizing their hyperparameters and outperform the other methods when predicting the next movies interested by users.

Key words: Deep learning, Long Short-Term Memory, Recommender systems, Sequence mining

School of Computer Science and Engineering, International University, Vietnam National University Ho Chi Minh City, Vietnam

Correspondence

Thi Thanh Sang Nguyen, School of Computer Science and Engineering, International University, Vietnam National University Ho Chi Minh City, Vietnam

Email: nttsang@hcmiu.edu.vn

History

- Received: 10-8-2019
- Accepted: 22-8-2019
- Published: 19-9-2019

DOI : 10.32508/stdjet.v3iSI1.540

INTRODUCTION

Nowadays, there is a huge of information on the Internet which leads to the difficulty of users for choosing the suitable one with their limitation time and thought. A lot of decisions must be given every day from the smallest to the biggest. Many topics have been given around the factor of making decisions, and a recommender system is the most successful strategy up to now. Many RS models depend on the relationship between users and items, but the one in the study relies on user sessions known as usage knowledge. One session of a user is the historical interactions of his/her on the items. In other words, one session of a user consists of one or more items clicked or rated by this person, so it can be described as the sequential data. Therefore, the deep recurrent network such as Recurrent Neural Networks (RNNs) and LSTM can demonstrate their effectiveness in processing this data. In this approach, a LSTM-based model is produced for building the movie recommender system relying on the user interaction data (session). Moreover, a comparison between the proposed approach versus RNN and a baseline method will be provided to find out the best model.

In the following, next section presents ***Related work***. Section ***Research Methodology*** gives a problem description and approaches to the LSTM-based movie recommender systems. Section ***Experimental results and evaluation*** proposes an optimization solution of the LSTM model for more effective movie recommendation. Discussion are explained in Section ***Discussion***. Section ***Conclusions*** concludes this study.

RELATED WORK

Recommender systems [1] play an important role in e-commerce systems, which help customers do transactions, such as, shopping or searching. Several popular approaches to recommendation are collaborative filtering models, content-based recommender systems, knowledge-based recommender systems, demographic recommender systems or hybrid and ensemble-based recommender systems. In this study, we consider the knowledge of user interactions with information systems to gain items of interest. In other words, this study aims to build knowledge-based recommender systems by mining user sessions. As known, user behavior is interested much in most recommender systems. Data mining algorithms which are often applied are sequence mining algorithms, such as, Apriori[2], tree-based or deep learning (DL). Tree-based algorithms have achieved high performance in terms of recommendation precision and saving memory for storing tree-based knowledge bases[3]. Recently, DL is a hot topic in this field since its advantages in terms of accuracy but takes much memory and time for training. According to A. Vieira[4], the buying session data which is extracted from the clickstream data of an e-commerce site can be learned using Deep Belief Networks and Stacked Denoising auto-Encoders. By experiments, it has been shown that the learned results can make purchase predictions with high accuracy, greater than 80%. In the study of R. Devooght, and H. Bersini[5], RNNs have been employed to the collaborative filtering process in order to make movie recommendations. The Movielens 1M and Netflix datasets were used, and it has been found that LSTM (a particular case of RNNs) overcomes the Markov chain model, the collaborative filtering using user-based nearest neighbors (user-KNN), and Bayesian Personalized Ranking – Matrix Factorization (BPR-MF).

As seen, RNNs are efficient for sequence mining in recommender systems. Therefore, this study focuses on RNNs and sequential pattern discovery, particularly, the user sessions of watching movies are discovered to build movie recommender systems. The following presents sequence model construction and deep recurrent neural networks for session-based movie recommender systems.

Sequence model construction for session-based movie recommender system

A session can be used as an ingredient to be fed under sequential data. In order words, a session is the list of movies rated by a specific user arranged in an order of timestamps, which is commonly used in sequences of mini batches. It is possible to use a sliding window as same as its application for words in sentences in natural language processing then put all windowed fragments next to each other in order to form mini-batches. However, that does not fit to the purpose of RS due to two reasons:

+ The length of the sessions is different between some users, it does not run steadily as same as the sentence structure sequence model: some sessions contain only one interaction (click, view, or rate) by one user, while others range more than hundreds.

+ The purpose for mining movie sessions is to capture how it evolves over a period, so splitting them into fragments makes no sense.

For solving the problem, the approach of session parallel mini-batches[6] is given. At the beginning, the arrangement for the sessions is created. After that, the first interaction of the first X sessions forms the input of the starting mini-batch (the ideal output become the second interactions of the active sessions). The next mini-batch's appearance is from the second interaction and so on. In reality, the system can understand when the session of user ended, from that state, the next available session will be put in place. In this model, sessions are assumed to be independent, so the appropriate hidden state is reset when this switch occurs. For more details, the session-parallel mini-batches foundation is described in Figure 1.

Deep recurrent network (LSTM) for session-based movie recommender system

LSTM - a straightforward solution of RNN is found by Sepp Hochreiter and Jürgen Schmidhuber in 1997[7] can tackle the vanishing and exploding gradient problem. The term "memory" is used instead of "neural" to perform the size and complexity in the structure of LSTM. The computational unit of the LSTM network is called the memory cell, memory block, or just cell for short, and there are many more computation tasks than the one in RNN.

In LSTM network, there are four main components including three gates, block input, memory cell, output activation function. For a complex structure of LSTM, each gate plays a specific role in computation:

+ Forget gate: chooses which information to discard from the cell.

+ Input gate: decides what values from the output to update the memory state.

+ Output gate: exposes the contents of the memory cell (or not) from the output of LSTM unit.

The output of the LSTM block is recurrently connected back to the block input and all of the gates for the LSTM block. The input, forget, and output gates in an LSTM unit have sigmoid activation functions for [0, 1] restriction. The LSTM block input and output activation function (usually) is a *tanh* activation function. As same as RNN or other Neural Networks, LSTM has both forward and backward process[8] to form the learning model.

RESEARCH METHODOLOGY

Problem description and approaches to the proposed LSTM-based movie recommender system

In this section, we present problems and approaches to the proposed LSTM-based movie recommender system.

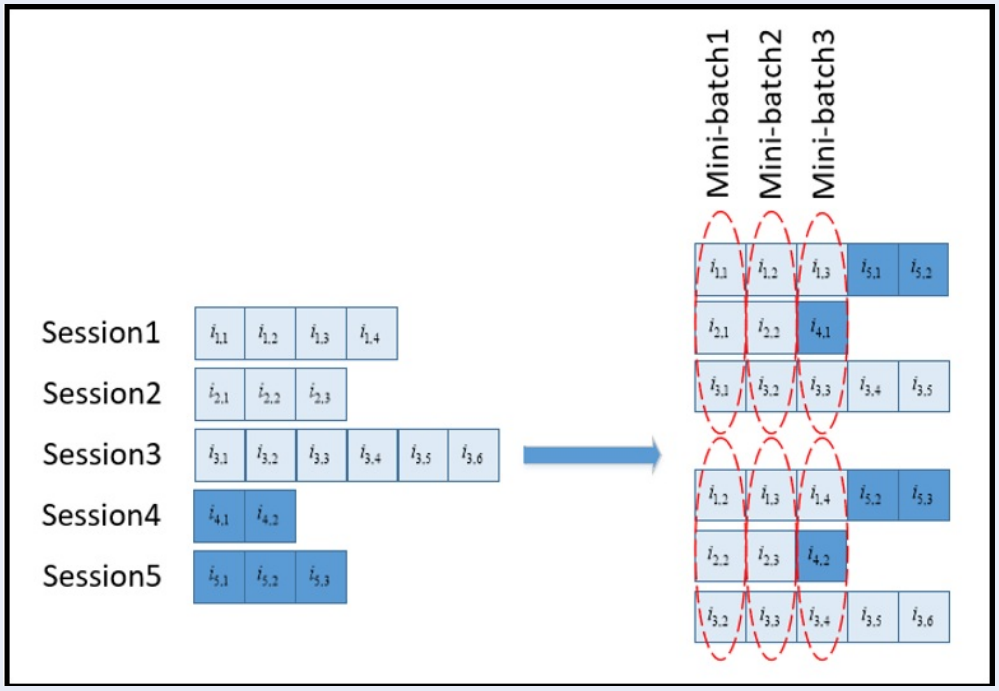

Figure 1: Session-based model consideration on Movie RS

Overall architecture of the LSTM-based movie recommender system

At the beginning, the input contains interaction data (sessions) of users under sequential type converted from a mini-batch model, which will be discussed briefly in the next section. The next task is to set up a presentation type for sequential data. In this approach, the sequence of sessions is zipped under Interaction, which is the numeric representation of users and items by timestamp. In general, a sequence interaction object consists of three factors:
+ The identifier of a user who made the interactions.
+ The tensor consists of several vectors of movie items, which are interacted by that user.
+ The maximum length of the sequence existed in the object.
After that, each sequence in the sequence interaction object is gone through an embedding layer, which is used to represent the weight of movie items followed by the order of the movie items in the sequence. The last item in the sequence receives the higher weight than others due to its up-to-date. Then, each embedded sequence is put through an amount of LSTM layers for learning. Finally, the system uses BPR[9] to produce the factorization matrix which is the predicted score on each movie. The sorted score can be used for recommendation. The overall architecture for this model is illustrated in Figure 2.

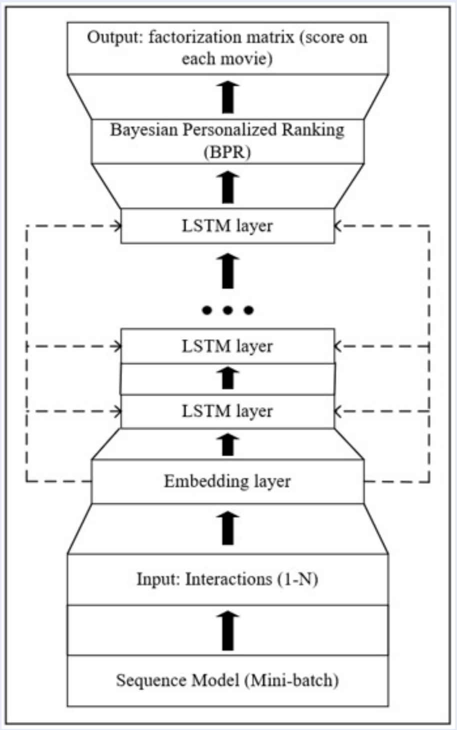
Output: factorization matrix (score on each movie)
Bayesian Personalized Ranking (BPR)
LSTM layer
LSTM layer
LSTM layer
Embedding layer
Input: Interactions (1-N)
Sequence Model (Mini-batch)

Figure 2: Overall workflow of the LSTM-based movie recommender system

Gradient Optimization for LSTM model in the movie recommender system

In this study, stochastic gradient descent (SGD) approach is used to optimize the gradient in the LSTM network due to its performance much faster in comparison with Mini-batch gradient descent. Specifically, the Adam [10] optimizer is applied in this study for LSTM networks. Adam is the abbreviation of adaptive moment estimation, which has a little bit different in comparison to classical SGD, it only needs first-order gradients thus there is an increment of performance in reality. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [10]. This methodology is the inheritance of the advantage of two other methods: AdaGrad [11] and RMSProp [12]. In facts, many DL models has used Adam instead of others, due to its ability to minimize cost. Adam keeps parameter's magnitude update is invariant to the rescaling of the gradient. Therefore, to keep Adam update efficiently, the choice of stepsize is compulsory. Therefore, there are several tests when defining a hyperparameter stepsize in order to return the best result of LSTM. More details, in Adam structure, four basic configuration parameters are mentioned as the learning rate α, two exponential decay rates β_1, β_2 in bounded [0, 1) and epsilon ε that is really small ($< 10^{-7}$) to avoid zero division.

Regularization methodology for the movie recommender system

The work of data handling in most of the data mining model is important. To handle data efficiently, there is one successful approach that is used widely in almost applications – dataset splitting. In this approach, the goal is whatever any group of splitting movie data is, the curve of the error function between them must fluctuate in an acceptable range, which is usually called appropriate fit.

More details, the data splitting approach consists of three sets as follows:

+ Training set: the part of data is applied in the task of building training set, it can be used to config some hyperparameters of the LSTM network such as the batch size, learning rate, the l2 loss penalty rate,… for checking regularization. In reality, one DL application has to be passed more than hundreds of hyperparameter configurations before publishing.

+ Validation set is usually used in model consideration and providing the frequent evaluation of the model in comparison to the training set.

+ Testing set is also used in evaluation. Specifically, unlike classification problem, evaluation in RS model plays the same role as unsupervised learning. In fact, there is no ensure that the list of movies recommended to user is correct or not. Thus, the used evaluation method is a statistical measure to visualize the distribution of the testing test and validation test.

Choosing a golden splitting ratio is an indispensable process in data splitting. It is usually called a cross validation process. The 80 – 20 or 90 – 10 (percentage) has been a golden ratio in both theoretical and practical applications. In this study, the 80% for training data, 10% for testing data and 10% for validation data are applied for cross validation.

Framework for detecting the appropriate data and evaluating the learning model in the movie recommender system

In this study, the MovieLens dataset [13] with approximately 10 million interactions is used. At the beginning, from the movie dataset, some features are chosen to build the proposed RS. Then, data preprocessing is applied to clean and convert user sessions to sequences. After that, the dataset splitting is applied to define three sets. The training set is fed into the model and do some tests to choose the best hyperparameter which can minimize the loss. Finally, the evaluation on the validation and testing sets by using Mean Reciprocal Rank (MRR) and other evaluation metrics as Precision@k, Recall@k and F1-Score@k is to confirm the used model is effective or not. The overall workflow of detecting the best model and the evaluation is shown in Figure 3.

In this study, Mean Reciprocal Rank (MRR) is used as a statistical method evaluating the model. In general, the Reciprocal Rank information retrieval measure calculates the reciprocal of the rank at which the first relevant item was retrieved [14]. When averaged across queries, the MRR is calculated. The formula of MRR is described as follows:

$$\mathrm{MRR} = \frac{1}{|\mathrm{Q}|}\sum_{\mathrm{i}=1}^{|\mathrm{Q}|}\frac{1}{\mathrm{rank_i}}$$

Where $rank_i$ refers to the rank position of the first relevant item in the i^{th} query; Q is the number of items. The model computes MRR score for both validation and testing. Then, the model is evaluated to be good when MRR scores given on both testing and validation set is approximately same. Other evaluation metrics used in this approach is Precision@k and Recall@k [14]. In comparison to MRR, these metrics care on k highest ranking items, which are the reasonable evaluation measures for emphasizing returning more relevant items earlier. The key point of this method is to take all the precisions at k for each the sequence in the testing set. More details, the sequence of length n is splitted into two parts: the sequence of length k for comparison and the other sequence of length $n-k$ put into the predicting function, and then the sorted factorization matrix scores are retrieved. If any item in top k highest scores matches the one in the sequence of length k, the number of hits is increased by 1. Then, the precision at k for one sequence of length n is given by the number of hits divided by k, which stands for the number of recommended items. For the recall, k stands for the number of relevant items. In facts, k in recall is usually smaller than the one in precision. Finally, the mean average precision and recall at k are calculated for all sequences in the testing set. In general, the formulas of the precision, recall and F1-score at k are described as follows.

$$Precision@k = \frac{relevant\_items\ in\ top\ k}{k}$$
$$Recall@k = \frac{relevant\_items\ in\ top\ k}{relevant\_items}$$
$$F1-score@k = 2 \times \frac{Precision@k \times Recall@k}{Precision@k + Recall@k}$$

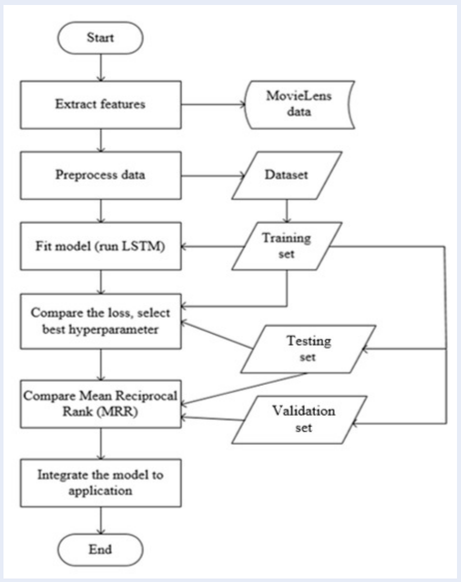

Figure 3: Overall workflow of detecting the best model for the movie recommender system

LSTM model Optimization

Hyperparameter optimization for LSTM model.

LSTM hyperparameter

There is not a good methodology to choose an ideal hyperparameter for the neural network up to now. Thus, the more trials, the better results are for the model. In this study, an automatically testing program with some random hyperparameters is built and takes three days consecutively to find the best hyperparameter. Some hyperparameters used in the configuration are embedding dimension, number of epochs, random state (shuffling number of interactions), learning rate, batch size... The loss function is kept same in the experiments.

Loss function

Several loss functions are applied to find the most appropriate model, the formulas of them are listed in Table 1.

Model efficiency evaluation

In this study, LSTM, RNN and another baseline method are chosen to compare the evaluation metrics. The common baseline is Item-KNN, which considers the similarity between the vectors of sessions. This baseline method is one of the most popular item-to-item solutions in practical systems. The MRR, Average Precision, Average Recall at 20 are measured to find out the efficiency of the LSTM versus the others.

EXPERIMENTAL RESULTS AND EVALUATION

Hyperparameter optimization

The experiment is taken by running 10 trials on the randomly selected hyperparameters which are defined in the fixed list as follows:

+ Learning rate: [0.001, 0.01, 0.1, 0.05]

+ l2: [1e-6, 1e-5, 0, 0.0001, 0.001]

+ Embedding dimension: [8, 32, 64, 128, 256]

+ Batch size: [8, 16, 32, 64, 128]

The loss function used in this experiment is BPR and the number of epochs for each trail equals 10. The loss sorted results is shown in Table 2.

The best result of the experiment is chosen for the model. In facts, the model requires a big system for training faster with more epochs, but the current one is not enough for training longer. Therefore, there should be more time for training a complete model.

Loss function

This experiment compares the efficiency of some loss functions mentioned in the previous section. The hyperparameter is chosen from the best one in the previous experiment. The results of the experiment on the training set in four types of loss after 10 epochs are illustrated in Figure 4.

Table 1: Loss function formulas for the model

Pointwise	$L=\sqrt{positive\_loss+negative\_loss}$ positive_loss = 1 - sigmoid(pos_pred) negative_loss=sigmoid(neg_pred)
Hinge	$L=\sqrt{max\{0,\ (neg\_pred\ -\ pos\_pred)\ +1\}}$
Adaptive Hinge	$L=\sqrt{max\{0,\ (neg\_pred\ -\ hightest\_pos\_pred)\ +1\}}$
Bayesian Personalized Ranking (BPR)	$L=\sqrt{1-sigmoid(pos\_pred\ -\ neg\_pred)}$

neg_pred: negative_prediction;
pos_pred: positive_prediction

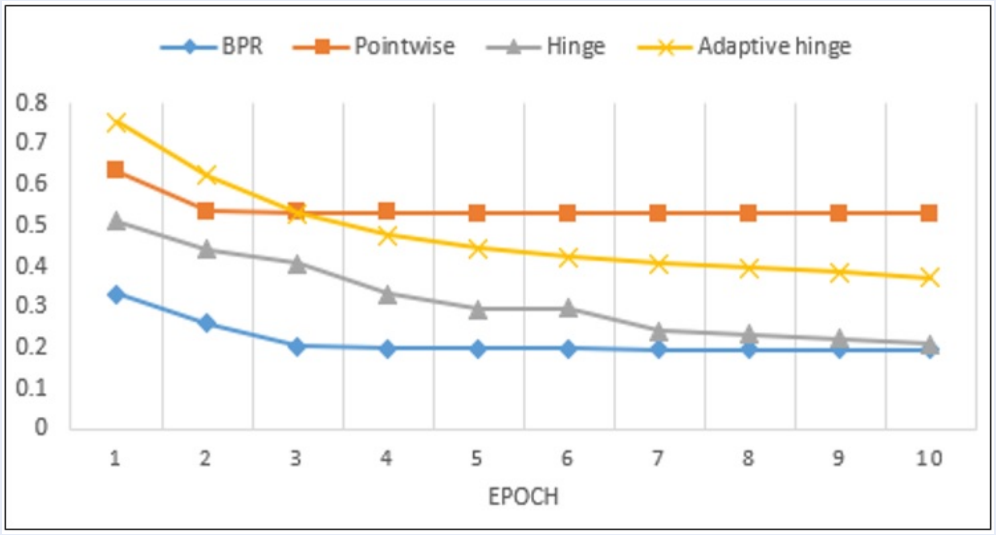

Figure 4: Loss function performance on the model

Table 2: Sorted loss results for several hyperparameters test

Batch size	Embedding dimension	Learning rate	L2	loss
128	64	0.001	0.0001	0.1962
64	64	0.01	0.0001	0.2491
8	8	0.01	1e-6	0.2537
16	256	0.1	1e-5	0.2701
16	32	0.05	0.001	0.271
16	128	0.01	0	0.276
16	64	0.001	1e-6	0.281
32	32	0.05	1e-5	0.2944
16	128	0.1	0.001	0.3566
8	128	0.05	0	0.4204

According to the graph in Figure 4, the BPR and Hinge loss can minimize the loss better than others can. Especially BPR, which can run the loss well as the beginning and it looks more stable than Hinge in regularization, as the comparison in training and testing in Figure 5. Therefore, BPR is chosen to perform the model instead of Hinge, Pointwise and Adaptive Hinge.

Evaluation results

The evaluation results between LSTM, RNN and the baseline Item-KNN method is shown in Table 3.
According to the experiment results, both LSTM and RNN can perform better than the baseline method (Item-KNN). Moreover, LTSM performs well in building a RS with specific relevant movies. LSTM can forget what it thinks to be not necessary for the long-term. Therefore, the learning results of LSTM are more updatable by time and frequency of interactions. Overall, LSTM is a better model for building the session-based RS.

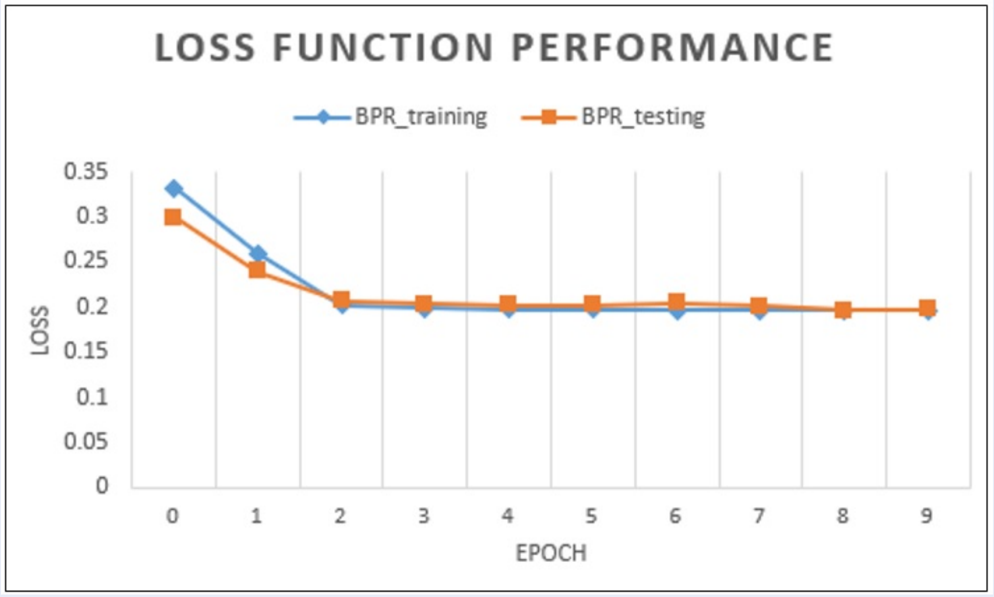

Figure 5: Good fit on training and testing by BPR loss function

Table 3: RNN on the testing set

DL Model	Precision @20	Recall @20	F1-Score@20	MRR (test/validation)
LSTM	0.705	0.707	0.706	0.239/0.235
RNN (dropout =0.2)	0.598	0.614	0.606	0.224/0.204
Item-KNN	0.506	0.507	0.506	0.201/0.206

DISCUSSION

In our approach, the modern model of recurrent neural networks, i.e., LSTM, is applied to do the movie RS with the task of session-based recommendations. Besides, the modification of LSTM in order to fit it better is performed by using session-parallel mini-batches and ranking losses. The evaluation results have shown the outstanding improvement in comparison with the popular baseline approach. The movie dataset provided by GroupLens is excellent for researching on some principal features. Thus, this approach can be applied not only for movie data, but also for some other practical fields.

CONCLUSIONS

In conclusions, the LSTM-based movie RS has been proposed and can achieve higher recommendation performance when optimizing the hyperparameters of LSTM, using loss function and optimization function. The loss is calculated to keep decreasing after each epoch and keep as minimum as possible for the long-term computation. Adam optimizer plays a great role in modifying the hyperparameter. Moreover, LSTM has been proved as the better model than RNN during evaluation. Despite the results of both are not good enough, but this study has presented some solutions to improve the accuracy of the learning model.

ACKNOWLEDGMENTS

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number: 06/2018/TN.

ABBREVIATIONS

RS: Recommender Systems
LSTM: Long Short-Term Memory
RNN: Recurrent Neural Networks
DL: Deep Learning
KNN: K-Nearest Neighbors
BPR-MF: Bayesian Personalized Ranking – Matrix Factorization

SGD: Stochastic Gradient Descent
MRR: Mean Reciprocal Rank

COMPETING INTERESTS

The authors hereby declare that there is no conflict of interest in the publication of the article.

AUTHORS' CONTRIBUTION

Duy Bao Tran is involved in proposing and implementing solutions, and writing reports.
Thi Thanh Sang Nguyen has giving ideas and solutions, assess experimental results and writing the manuscript.

REFERENCES

1. Aggarwal CC. Recommender Systems. Springer International Publishing. 2016;Available from: https://doi.org/10.1007/978-3-319-29659-3.
2. Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases. Proceedings of the 20th International Conference on Very Large Data Bases. 1994;p. 487–499.
3. Nguyen TTS, Lu H, Tran TP, Lu J. Investigation of Sequential Pattern Mining Techniques for Web Recommendation. International Journal of Information and Decision Sciences (IJIDS). 2012;p. 293–312. Available from: https://doi.org/10.1504/IJIDS.2012.050378.
4. Vieira A. Predicting online user behaviour using deep learning algorithms. arXiv:151106247v3. 2016;.
5. Devooght R, Bersini H. Collaborative Filtering with Recurrent Neural Networks. arXiv:160807400. 2016;.
6. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based Recommendations with Recurrent Neural Networks. 2015;.
7. Graves A. Supervised Sequence Labelling with Recurrent Neural Networks. Springer-Verlag Berlin Heidelberg. 2012;Available from: https://doi.org/10.1007/978-3-642-24797-2.
8. Patterson J, Gibson A. Deep Learning-A Practitioner's Approach. O'Reilly Media, Inc. 2017;.
9. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Canada,. 2009;p. 452–461.
10. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2015;.
11. Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. J Mach Learn Res. 2011;12:2121–2159.
12. Hinton G, Srivastava N, Swersky K. Lecture 6d- a separate, adaptive learning rate for each connection. Slides of Lecture Neural Networks for Machine Learning. 2012;.
13. Grouplens. 2019;Available from: https://grouplens.org/datasets/movielens/.
14. Liu L, Ozsu MT. Encyclopedia of Database Systems. Springer. 2009;Available from: https://doi.org/10.1007/978-0-387-39940-9.

Tạp chí Phát triển Khoa học và Công nghệ – Engineering and Technology, 3(S1):SI1-SI9

Open Access Full Text Article

Bài Nghiên cứu

Đề xuất phim dựa trên bộ nhớ ngắn hạn

Trần Duy Bảo, Nguyễn Thị Thanh Sang*

TÓM TẮT

Hiện nay, các hệ thống đề xuất đã trở thành một công cụ cơ bản để giúp người dùng đưa ra quyết định trong hàng triệu lựa chọn khác nhau - kỷ nguyên của Dữ liệu lớn. Nó mang lại lợi ích rất lớn cho nhiều mô hình kinh doanh trên toàn thế giới do hiệu quả của chúng đối với khách hàng. Rất nhiều mô hình và kỹ thuật khuyến nghị đã được đề xuất và có nhiều kết quả đáng kinh ngạc. Phương pháp lọc cộng tác và phương pháp lọc dựa trên nội dung là phổ biến, nhưng cả hai đều có một số nhược điểm. Một điều quan trọng là chúng chỉ tập trung vào sở thích tĩnh dài hạn của một người dùng trong khi bỏ qua các mẫu giao dịch ngắn hạn, dẫn đến bỏ sót sự thay đổi sở thích của người dùng trong suốt thời gian. Trong trường hợp này, mối quan tâm của người dùng ở một thời điểm nhất định có thể dễ dàng che mờ bởi các hành vi quyết định trong lịch sử của người đó, dẫn đến các khuyến nghị không đáng tin cậy. Để giải quyết vấn đề này, một phiên tương tác của người dùng với các mục có thể được coi là một giải pháp. Trong nghiên cứu này, các mạng Bộ nhớ ngắn hạn (LSTM) sẽ được phân tích để áp dụng cho các phiên của người dùng trong hệ thống đề xuất. Bộ dữ liệu MovieLens được dùng trong nghiên cứu hệ thống tư vấn phim. Một số thí nghiệm được thực hiện để đánh giá hệ thống đề xuất phim dựa trên LSTM.

Từ khoá: Học sâu, Bộ nhớ ngắn hạn, Hệ thống đề xuất, Khai thác chuỗi

Khoa Công nghệ Thông tin, Trường Đại học Quốc tế, ĐHQG-HCM, Việt Nam

Liên hệ

Nguyễn Thị Thanh Sang, Khoa Công nghệ Thông tin, Trường Đại học Quốc tế, ĐHQG-HCM, Việt Nam

Email: nttsang@hcmiu.edu.vn

Lịch sử

- Ngày nhận: 10-8-2019
- Ngày chấp nhận: 22-8-2019
- Ngày đăng: 19-9-2019

DOI : 10.32508/stdjet.v3iSI1.540