

## A Product Recommendation Model Based on Recurrent Neural Network

Naresh Nelaturi\*, Golagani Lavanya Devi

Department of computer science and systems engineering, College of Engineering(A), Andhra University, Visakhapatnam-530003, India

Corresponding Author Email: [naresh855@gmail.com](mailto:naresh855@gmail.com)

<https://doi.org/10.18280/jesa.520509>

### ABSTRACT

**Received:** 13 July 2019

**Accepted:** 27 September 2019

#### Keywords:

recurrent neural network (RNN),  
purchase patterns, deep learning,  
bidirectional model, attention  
mechanism

In traditional recommender systems, the product recommendations are generally made based on the static behavior or preference of customers. This paper designs a novel production recommendation model that processes customer-product interaction data as a time-based sequential data, and makes personalized product recommendations based on the purchase patterns of customers. Specifically, the model relies on the deep learning technique of recurrent neural network (RNN) to uncover the dynamics in purchase patterns of customers; a bidirectional model with attention mechanism was introduced to personalize the product recommendations. The effectiveness of the proposed model was verified through an experiment on a benchmark dataset called Movie lens. The experimental results show that the RNN-based model can efficiently capture the temporal dynamics of customer preferences, and then generate highly individualized product recommendations.

## 1. INTRODUCTION

Recommender systems seeks to render likely interested products of customers based on their preferences. Increasing information overload and interests of web users made recommender systems a prominent tool that redefined the way to access potential information on web with ease. Ecommerce giants like Amazon, Netflix, Flipkart and Alibaba made their web systems equipped with recommender systems to attain customer satisfaction which led to increase in their profits. The task of recommending products to customers can be formulated as a rating prediction problem or as a product ranking problem. Approaches to solve these problems utilize customers' characteristics data, products' features and customer-item interaction data. Generally, methodology for rating prediction problem come from three paradigms: collaborative, content-based and hybrid. However, several studies have shown that rating prediction is not efficient in generating top-n recommendations. Alternatively, nowadays research in recommender systems is shifted more towards ranking task. Formulating a recommendation problem as a personalized ranking problem with an objective of designing a model for sequencing of products according to customers' preferences. However, such a modelling on interaction data may not yield fruitful personalization as it captures only static behavior of customers. Customers intuition on a product is not consistent, there can be dynamic drift in his interests caused by various latent factors such as seasonal changes, perception on product and social group influence. Many of the existing approaches are not leveraging potential information about customers intuition of items over time. To generate effective personalized recommendations the customer dynamic behavior is to be characterized and interpreted from his interactions. An approach for exploiting temporal dynamics by considering sequence properties in the customer product interaction data can be helpful in rendering

augmented recommendations. Models leveraging deep learning techniques has achieved noteworthy success in processing sequential data. In particular, recurrent neural networks gained popularity in tasks involving sequential learning. Wide range of applications where RNN are predominantly used and produced produces state of the art performance include biological sequence analysis, activity recognition, machine translation, speech recognition, music generation and sentimental analysis [1-4].

This paper seeks to generate quality product recommendations by design a deep learning-based model for capturing temporal dynamics and sequence structures in the interaction data. The hypothesis of this model is given a sequence data of customer-item interactions along with items' features, the model will discover and capture the dynamics of purchase behavior patterns and predict the next sequence of items that are most likely to be purchased by a customer. The intuition is that extraction of temporal sequential features from purchase history can uncover the dynamic representations of unexpressed customer preferences. Such a model can be trained by feeding session-wise customer interactions to recognize sequential purchase patterns of customers across sessions. For prediction when the model is presented with most recent customers' purchase data of latest/current session, it identifies the patterns with the past sequential features and generates efficient and relevant recommendations.

*Naïve solution:* A naive solution is designing an RNN architecture with an objective of extracting and capturing temporal sequential features and dependencies from interaction data(ratings). In general, recurrent networks are inhabited with an exclusive problem of not learning dependencies for longer sequences.

Along with this, standard RNNs are forward networks having unidirectional units. Such a network suffers from the limitation of not exploiting the dependencies of patterns

ahead of the present unit. During training, processing of large varying sequential data may cause hidden units of the model to hold noisy information. Further, transition of this noise to subsequent layers degrades the performance of the model.

*Augmented/Better solution:* This work utilizes RNN with gated cells as hidden units to combat problem of vanishing gradient. Further, the proposed model seeks remedy for exploitation of unseen pattern dependencies by considering a bi-directional RNNs. Such a recurrent model in a given context exploits not only past information but also future information. The model considers fixed length padded sequences and examines the important timesteps. Further enabling model with attention mechanism allows to attain a summary of customers preference and intents in a session. Contributions made by this study in modelling temporal dynamics of customers purchase to augment product recommendations are:

- The work provides an insight in generating augmented product recommendations by modelling customer's temporal behavior and learning sequential properties in product purchase from interaction data.
- The proposed variant of recurrent neural network model is a multi-layered bi-directional network with attention mechanism to summaries and understand the intention across customer interacting session. Potential products are considered for prediction using output sampling
- Extensive experimentation is carried out with wide range of metrics for evaluating from multiple perspectives. Comparative evaluation is performed with baseline and state of the art recommender methods. Research findings of the experimentation are discussed.

The paper is structured as follows: Section 2 briefs on concepts of recurrent neural networks and its implication in the area of recommender systems. Section 3 proposes an advanced recurrent neural network architecture for modelling purchase patterns and generate effective personalized recommendations.

Section 4 discusses the empirical evaluation carried out to validate proposed approach and presents results obtained. Lastly, Section 5 summarizes the study and suggests future extensions for improving performance of the models.

## 2. RELATED WORK

This section provides a review on deep learning-based approaches for processing sequential interaction data using recurrent neural network for the tasks of recommender systems.

Over the past years the factor models and neighborhood techniques have been the state of art methods for recommender systems [5-6]. However, these methods are inefficient in exploiting sequential properties in the interaction data. Consequently, the behavior of the customer is not considered in rendering recommendations. This is due to the fact that full dimensional representation of the ordered sequence is provided for the model. A few studies were carried out using customers' interactions with the system as a sequential data for capturing and understanding patterns in the purchase behavior. Supervised sequential learning techniques such as association mining, markov and its variants, sliding and recurrent sliding window models are successfully leveraged to solve the sequential learning problems [7-8].

However, most of the real-world recommender problems involved in utilization of sparse data which is not viable for applying aforementioned learning techniques directly and also the techniques when applied for recommender problems result high computational complexity [9]. On the other end, researchers' interest over the deep learning-based approaches for recommender system gained momentum in the past few years. In specific, to perform sequential modelling for session-aware/session-based recommendations the recurrent neural networks and its variants are being explored. In modelling purchase patterns of customers to generate recommendations effectively this work/study leverages recurrent neural networks using session wise interaction data. In this regard an insight in RNNs and literature of recurrent neural network-based recommender systems is as follows:

### 2.1 Recurrent neural networks

Recurrent neural networks (RNN) are the most popular member among the family of deep learning techniques used extensively for processing text, audio and video data. The rise in the popularity is due to the state of the results attained in processing complex sequential structured data. Unlike any other technique in deep learning family, RNN is special kind of neural network having memory to remember previous computations. These networks have internal looping mechanism that gives the ability to learn from previous states. A breakthrough in recurrent neural network architectures came with the addition of new models to the RNN family. This happened with introduction of long short-term memory (LSTM) and gated recurrent unit (GRU) in the RNN architectures. These models equipped with special memory cells in recurrent units that have alleviated problem of vanishing gradients [10-11]. The learning encompassing update of weights of the network to reduce error.

Backpropagation Through Time (BPTT) algorithm is employed as training algorithm to calculate gradients of the cost function by unfolding all the time steps [12]. Recent advances and extensions in deep recurrent neural networks adopted bidirectional, stacking and attention mechanisms in its network architecture [13-14]. These augmented networks have produced significant outbreaks in machine translation, video/image tagging and in many other domains. Bidirectional recurrent network embraces forward and backward directed layers, therefore leverages information and learns representations from future and past dependencies. Stacking the recurrent network with multi layers causes an increase in amounts of abstraction in multiple perspectives. Consequently, allowing the recurrent network to have better representations of sequential properties. Further, most recent studies using deep neural networks have adopted attention mechanism in recurrent architecture to provide high quality representation with an emphasis on relevancy of the content to the context.

### 2.2 Recurrent neural network-based recommenders

The superiority of matrix factorization techniques in solving recommender problems to some extent is challenged by the RNN based approaches and is an active area of research. A comprehensive survey with deep insight's for developing deep learning-based approaches and techniques for recommender systems is given by Zhang et al. [15]. Fakhfakh et al. has presented advances in deep learning-

based recommendations with emphasis on current issues and challenges [16]. In the scenario of sequence modelling and session-based recommendations Hidasi et al. [17] have proposed Gated Recurrent Units based approach. Liu et al. [18] proposed RNN approach for temporal learning to job recommendations. Hidasi and Alexandros has proposed RNN with Top-K for sequential data [19]. Elena and Flavian has taken into account contextual information in modelling behavior and presented Contextual RNN [20]. Tang et al. has utilized bi-directional RNN for recommending movies [21]. RNN based recommendation approach have shown robustness to data sparsity and have the ability to learn short-term as well as long-term dependencies. Consequently, an RNN based approach to model the dynamics of customer purchase patterns to augment quality of personalized recommendations is presented in this work.

### 3. THE MODEL

This section presents our proposed RNN architecture for predicting next sequence of likelihood items that are of customers interests. Predicting customers' interest in a session by analyzing and capturing dynamic purchase pattern of customers to generate relevant product recommendations is a challenging problem. This problem has been considered as a potential problem in industry and has attracted many practitioners and researchers in recent times. Despite a few solutions being proposed in recent years by extending traditional recommender models, lack of standardized approaches and benchmark datasets made session-based recommendations still a viable problem. Due to the fact that recurrent neural networks are well known for handling

Design of the model with emphasis on motivation of its layers is discussed in this section. Finally, details of hyperparameters optimization by fine tuning is provided.

**Problem formulation:** In the scenario of generating top- $n$  personalized recommendations for a customer  $C_i \in C$  interacts with the system to browse/purchase products over a number of sessions (varying time intervals)  $S = \{S_1, S_2, \dots, S_t\}$ , each interaction  $S_t$  is described as a series of products  $S_t = (x_1, x_2, x_3, \dots, x_j, \dots, x_n)$  where  $j \in P, x \in R$ . The model considers the purchase behavior of customer in session wise and learns the sequential properties in the purchase pattern. The model then generates a list of products for a customer on ground of most likelihood products that will be of interest in the next interaction. Therefore, this problem is formulated as a sequence to sequence supervised learning problem and a deep recurrent neural based architecture is proposed to solve the problem. Where the notations used are described below:

- P: Set of products available in the catalogue.
- C: Set of customers of the system.
- t: time index in a session.
- i: customer index.
- j: product index.

#### 3.1 Design of the model

The architecture designed for aforementioned problem is a multi-layered neural network comprising of bidirectional recurrent layers with attention layers stacked on the top of it. The design of the proposed approach is illustrated as a computational graph comprising of individual modules is shown in Figure 1. Pre-processing of sequences is done to

produce fixed length sequence by padding. The proposed multi-layered architecture is composed of various hidden layers grouped into four modules. Given a session interaction of customer a  $C_i$  interaction with items over a particular period (session k) the model has to predict next sequence of  $n$ -items that may be of customers interests.

A brief outline of these modules and their motivation in the architecture is as follows:

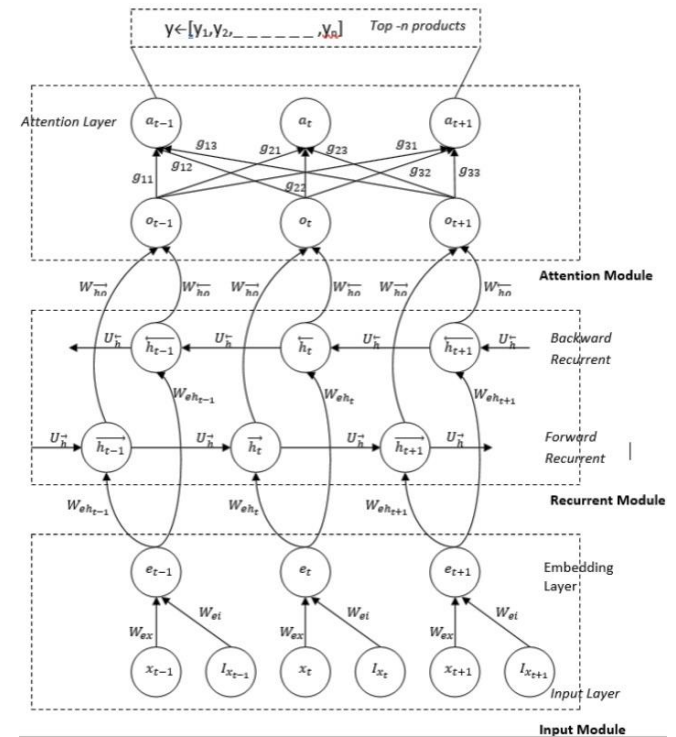


Figure 1. RNN Architecture to model purchase patterns

#### 3.2 Design of the input module

Objective of this module is to generate efficient representation of customer session and to facilitate incorporation of auxiliary information about items perused in the session. This module has two layers: input and embedding.

**Input layer** takes session-wise customers' product interaction data and generates a fixed length vector representation of session information. This vector passed to embedding layer that integrates with auxiliary information of items extracted from any pretrained convolutional neural network architecture. A non-linear function, ReLU is utilized for activation of hidden units, that also concatenates session data with auxiliary information. This module generates rich high dimensional representation of customer preferences in a session as vector  $e = [e_1, e_2, e_3, \dots, e_l]$ . The neuron units in the embedding layer bilinear units which perform bilinear mapping of rating and image feature of the products.

$$e_t = f_e(W_{ei} \cdot I_x + W_{ex} \cdot x_t + b_e)$$

where,  $f_e$  is a nonlinear function tanh is used for performing projecting the integration information to a new dimensionality.  $x_t$  and  $I_x$  represents vectors about the customer preference on a products and products feature map, respectively.  $b_e$  indicates the bias term.

### 3.3 Design of the recurrent module

Objective here is to learn higher level representations of dynamic customer preferences by extracting information from product images into a embedding layer vector. Recurrent module captures the temporal dependencies in the session-wise customer purchase history. The proposed model utilizes bidirectional recurrent layer comprising of two layers. One layer for processing sequence in forward direction i.e. dependencies of past on present or recent session. Its successive layer process data in backward direction. During the training the model has to estimate the parameters  $W_{eht}$  over the edges from embedding layer to recurrent layer. Empowering bidirectional mechanism to the hidden layers increase the utility of the dynamic representations in the subsequent layers. Further, model has to learn the parameters  $U_{h\rightarrow}$  and  $U_{h\leftarrow}$  of both the recurrent layers. The unit in the recurrent layers holds the state that is the representation of the sequence up to the time period t.

Hidden state ( $h_t^{\rightarrow}$ ) is updated in the forward directed recurrent layer as follows:  $h_t^{\rightarrow} = f_{hf}(W_{eht} \cdot e + U_{h\rightarrow} \cdot h_{t-1} + b^{\rightarrow})$

Hidden state ( $h_t^{\leftarrow}$ ) is updated in the backward directed recurrent layers using the following equation:

$$h_t^{\leftarrow} = f_{hb}(W_{eht} \cdot e + U_{h\leftarrow} \cdot h_t + 1 + b^{\leftarrow})$$

Here  $f_{hf}, f_{hb}$  are the non-linear transformation functions for activation of the units and  $b^{\rightarrow}, b^{\leftarrow}$  are the bias. The layers in the recurrent layers can be stacked to attain high dimensional representation of the purchase patterns.

### 3.4 Design of the attention module

Objective of this module is to improve the interpretability of information learned from the bi-directional recurrent module by utilizing temporal attention mechanism.

This module has two layers to compute scores that determine the interest and intention for a product in a session. The first layer aggregates the output of forward ( $h_t^{\rightarrow}$ ) and backward ( $h_t^{\leftarrow}$ ) recurrent layers in its units ( $o_t$ ) as follows:

$$o_t = f_o(W_{ho}^{\rightarrow} \cdot h_t^{\rightarrow} + W_{ho}^{\leftarrow} \cdot h_t^{\leftarrow} + b_o)$$

where,  $f_o$ , is an activation function that maps concatenated outputs to a non-linear boundary.  $W_{ho}^{\rightarrow}, W_{ho}^{\leftarrow}$  represents the weights associated with bidirected recurrent layers.  $b_o$  denotes the bias in this layer. The values in the units of this layer are projected into the attention layer to determine the score that indicates the customers intention on the product at time t ( $a_t$ ). this is performed utilizing the following equation:

$$a_t = f_a(gT \cdot o_t)$$

Here  $g$  represents the weight matrix,  $f_a$  is an activation function (ReLU) that generates a scalar value.

### 3.5 Design of the output module

Objective of this module is to predict the next sequence of items that can be selected/purchased by the customer. This module acts as a predictive layer that predicts sequence of products that a customer a peruse in this next interaction. Loss function is the objective function Bayesian personalized

ranking function that has to be optimized during training by the optimizer algorithm.

Generating top-n personalized recommendations from large number of products available in catalogue is not viable for any technique. To have an efficient computational feasibility the model is to be feed with a proper potential sample is to be obtained. This study considered popularity-based sampling technique for output sampling. This network of model is parameterized with the following hyper parameters: sequence length, embedding dimensions, number of hidden layers in the recurrent module, number of hidden units/ neurons and their activation functions in recurrent module, attention dimension, attention type, learning rate (alpha) in back-propagation, dropout/regularization.

Choice and initialization of these hyper-parameters is done before the commencement of learning process. These values are independent of data and selection does not depend on the algorithm being utilized. However, hyper-parameters initialization is of utmost important as they guide the process of determining final values for parameters of the model. Consequently, to alleviate the impact of generalization errors of the model, these hyper-parameters are to be tuned for improving model performance by running numerous experiments.

This proposed RNN model can generate personalized recommendation to a customer more precisely, with the following advantages. This approach leverages time line of customers' purchase patterns in past to present and future to present. By exploiting the sequences in both directions improves the representational power of the network. Attention mechanism determine level of intention the customer has on a product at in a session. Which enables the model to better summarize purchase behavior in the sessions. Also, model has Improved memorization capabilities for long sequences with attention mechanism. Such an RNN model can have stability and cab also be pretrained.

## 4. EXPERIMENTS AND RESULTS

For the purpose of evaluating the proposed model the requirements are: rating *dataset* with timestamps providing the ordered sequence of customer interactions, *methodology* specifying the procedure for training the model and conducting offline experiments, item ranking *metrics*, results and discussion on the experimental analysis that support the hypothesis of the study. The details of these components are given below.

### 4.1 Dataset details

Experiments are carried out on MovieLens dataset having a data of format < customer, movie, rating, timestamp >. Dataset hosted by Group Lens research project is the most widely utilized state-of-the-art dataset in the literature of recommender systems [http://www.movielens.org]. The dataset contains 1 million ratings of 3,883 movies given by 6,039 customers. Ratings given by the customer are scaled on 1 to 5. The interaction matrix having a density of 4.3 % making it more viable for experimental analysis than any other recommender datasets. Further, dataset provides timestamps details when a customer has rated a movie which enables to define sessions of interactions by the customers. Specifically, MoveLens1M dataset has around 163 average

interaction per customer and 289 interactions per item. A movie is characterized by its genres, theme, director and cast. This dataset has 3,883 movies belonging to 18 genres. A Movie poster provides abstract information about the movie characteristics. All the metadata about movies is not provided in original dataset but can be scraped from the Movie Database (<https://www.themoviedb.org/>) using feature *movieid* of dataset. The rationale for considering Movielens1M dataset as a primary dataset for the evaluation of the model is that it has the advantages of having good density, timestamps of ratings and posters. The goal of the model is to predict the next sequence of products (movies) that are most likely interestingness for a Customer.

## 4.2 Evaluation procedure

The task of the proposed RNN model in this experiment is to predict and rank the next sequence of movies that are of interest for a customer. Evaluation procedure adopted here comprises of partitioning of dataset, comparative analysis with state-of-art models and metrics. In order to perform sequential supervised learning, the dataset considered is to be partitioned into training and test dataset. The method adopted for partitioning in this experiment is sliding window protocol. This method partitions the sequence in data in several sample chunks of equal width for training and subsequent sequence for test. Advantage of this method is that it allows data to be used by any sequential learning algorithm. In evaluating the model performance for a customer average of each sample error is computed.

After partitioning of data into train and test sets, model is trained with training data and parameters of the model are estimated. Initialization of network weights is to be commenced before the training of model begins. Random sample of weights are generated using Gaussian distribution model with mean and standard deviation set to 0 and 0.1 values respectively. Number of neurons in each of hidden layer in recurrent modules is set 100. Hyperparameters of the model such as mini-batch, dropout, momentum and learning rate are optimized by running the experiment 50 times. To train the model on data in this experiment, implementation is done in *TensorFlow* with popular optimizers ADAM and RMSProp [22-23].

### 4.2.1 Methods for comparative analysis

To investigate the quality of the personalised recommendations for the next session of a customer the proposed model is compared with four popular recommender models. Models leveraging factorization and personalised techniques are used along with baseline for comparative evaluation are as follows.

- *POP*: A non-personalised technique that recommends the most popular products in the system to customers. Products' popularity based on the total number of interactions of the customer [24].
- *SVD/MF*: Singular Value Decomposition is one of the most successful matrix factorization approach for recommender systems. This approach generates latent factors dimensional space using customer and product characteristics. When predicting the likely interested products for a customer this space is used [25].
- *BPR-MF*: Bayesian Personalized Ranking is a latent factorization method for learning to rank products. BPR-MF

utilizes pairwise ranking approach to efficiently rank the personalized products for a customer [26].

- *Item-KNN*: Neighbourhood based approaches are the most popular and widely used along with MF techniques. Here, model seeks to find the items which are nearest neighbours to the items already purchased by the customers [27].

### 4.2.2 Evaluation metrics

The metrics adopted in this evaluation procedure are of type relevance and ranking metrics. Intuition of the study is proposed model generates next sequence of product reflecting customer preferences. These metrics measure how far the model is good at recommending relevant products and also its relative positioning in the list. Metrics for measuring the Performance of the recommender models are, Recall, Precision (Pr) and Mean Average Precision (MAP) [28].

Recall@n determines ratio of preferred /relevant products in the list of size 'n' to the total number of preferred products. Usually, recall correlates with the metric hit rate and click through rate. Precision@n determine ratio of relevant products recommended in list of size 'n'. Both recall and precision emphasize on the relevance recommendation in list and are not concerned about the position of products in the list. Mean Average Precision (MAP) is the most popular metric for measuring accuracy in ranking of products in the recommendation list. MAP provides single value summary of effective position of products in the rendered recommenders list.

## 4.3 Results

Table 1 summarizes the recommendation performance of proposed RNN and four methods considered on the Movie lens 1M dataset. Some of the interesting findings from the empirical evaluation carried out are listed here. Proposed RNN is the best performing model in terms of predicting the relevant products for a customer. Further, the ranking quality of RNN is also better in comparison to the other models. Among the other models, BPR-MF is poor in predicting in comparison to the other personalized models. With optimized parameters of item KNN it has shown better performance over non-personalized models. Popularity based methods are the ready to go methods for many recommender systems. Less computation requirement for training and testing makes them popular. But often these methods produce poor performance. Matrix factorization techniques are known to perform well when the density of the dataset is good. But to improve the performance of MF models, it has many model parameters are to be adjusted causing computationally complex than other techniques.

Performance metrics of the Item-KNN model are competitive after adjusting neighbors in model to 200. Further, Item-KNN being a memory-based model requires large memory requirements for computations. These models perform instance learning and does not need larger amount of training time. However, testing usually takes more amount of time compared to other models. As expected, the proposed approach delivered impressive results with considered metrics. Its is able to recommend relevant products as per customers' preferences as observed with metrics Recall and Precision.

**Table 1.** Top-n recommendation on movie lens dataset

| Method       | Metrics |          |           |       |       |
|--------------|---------|----------|-----------|-------|-------|
|              | MAP     | Recall@5 | Recall@10 | Pr@5  | Pr@10 |
| POP          | 0.067   | 0.021    | 0.037     | 0.111 | 0.101 |
| MF           | 0.095   | 0.039    | 0.059     | 0.152 | 0.135 |
| KNN          | 0.131   | 0.061    | 0.105     | 0.198 | 0.182 |
| BPR          | 0.090   | 0.032    | 0.056     | 0.128 | 0.120 |
| Proposed RNN | 0.148   | 0.063    | 0.111     | 0.216 | 0.198 |

## 5. CONCLUSION AND FUTURE WORK

This paper conducted a study to determine the effect of temporal dynamics in customer purchase patterns in generating relevant personalized recommendations. Accordingly, a bi-directional recurrent neural network with attention mechanism is proposed for modelling temporal structures in customer-product interaction data. The results from experiments suggest that modelling sequence modelling of customers' purchase patterns is useful in terms of understanding changing preferences and leads to rendering of augmented recommendations. The effectiveness of bidirectional and attention mechanism in of customers' is observed in the empirical evaluation. Future works is needed to address the feeding of varying length session data to the model. Further investigations are needed for efficient alternative sampling of products in the output layer from total products in the catalogue. A study investigating the employment of multi linear units in the recurrent component for integrating many meta-features of the products to improve the representation of the product and its customer preference.

## ACKNOWLEDGMENT

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India being implemented by Digital India Corporation (formerly Media Lab Asia).

## REFERENCES

[1] Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(1-2): 305-307. <https://doi.org/10.1007/s10710-017-9314-z>

[2] Graves, A., Mohamed, A.R., Hinton, G. (2013). Speech recognition with deep recurrent neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing. <https://doi.org/10.1109/icassp.2013.6638947>.

[3] Arevian, G. (2007). Recurrent neural networks for robust real-world text classification. IEEE/WIC/ACM International Conference on Web Intelligence (WI'07). <http://dx.doi.org/10.1109/wi.2007.126>

[4] Goel, K., Vohra, R., Sahoo, J.K. (2014). Polyphonic music generation by modeling temporal dependencies using an RNN-DBN. Lecture Notes in Computer

Science, Springer International Publishing, 217-224. [http://dx.doi.org/10.1007/978-3-319-11179-7\\_28](http://dx.doi.org/10.1007/978-3-319-11179-7_28)

[5] Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Transactions on Knowledge and Data Engineering, Institute of Electrical and Electronics Engineers (IEEE), 17(6): 734-749. <http://dx.doi.org/10.1109/tkde.2005.99>

[6] Ricci, F., Rokach, L., Shapira, B. (2015). Recommender systems: Introduction and challenges. Recommender Systems Handbook, 1-34. [http://dx.doi.org/10.1007/978-1-4899-7637-6\\_1](http://dx.doi.org/10.1007/978-1-4899-7637-6_1)

[7] Mandic, D.P., Su, L.G., Aihara, K. (2005). Sequential data fusion via vector spaces: Complex modular neural network approach. Workshop on Machine Learning for Signal Processing, IEEE. <http://dx.doi.org/10.1109/mlsp.2005.1532890>

[8] Witten, I.H., Frank, E., Hall, M.A., Pal, C.J. (2016). Data mining: Practical machine learning tools and techniques. Morgan Kaufmann. <http://dx.doi.org/10.1016/b978-0-12-374856-0.00026-2>

[9] Baltrunas, L., Ludwig, B., Ricci, F. (2011). Matrix factorization techniques for context aware recommendation. Proceedings of the fifth ACM Conference on Recommender Systems - RecSys'11, ACM Press. <http://dx.doi.org/10.1145/2043932.2043988>

[10] Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence [Internet]. Institute of Electrical and Electronics Engineers (IEEE), 31(5): 855-68. <http://dx.doi.org/10.1109/tpami.2008.137>

[11] Tjandra, A., Sakti, S., Manurung, R., Adriani, M., Nakamura, S. (2016). Gated recurrent neural tensor network. 2016 International Joint Conference on Neural Networks (IJCNN) IEEE. <http://dx.doi.org/10.1109/ijcnn.2016.7727233>

[12] Ahmad, A.M., Ismail, S., Samaon, D.F. (2004). Recurrent neural network with backpropagation through time for speech recognition. IEEE International Symposium on Communications and Information Technology, IEEE. <http://dx.doi.org/10.1109/iscit.2004.1412458>

[13] Graves, A., Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5-6): 602-610. <http://dx.doi.org/10.1016/j.neunet.2005.06.042>

[14] Vinyals, O., Toshev, A., Bengio, S., Erhan, D. (2015). Show and tell: A neural image caption generator. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA. <http://dx.doi.org/10.1109/cvpr.2015.7298935>

[15] Zhang, S., Yao, L., Sun, A.X., Tay, Y. (2017). Deep learning-based recommender system: A survey and new perspectives. ACM Comput, 52(1). <https://doi.org/10.1145/3285029>

[16] Fakhfakh, R., Anis, B.A., Amar, C.B. (2017). Deep learning-based recommendation: Current issues and challenges. International Journal of Advanced Computer Science and Applications, The Science and Information Organization, 8(12).

- <http://dx.doi.org/10.14569/ijacsa.2017.081209>
- [17] Tan, Y.K., Xu, X., Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems - DLRS 2016. ACM Press. <http://dx.doi.org/10.1145/2988450.2988452>
- [18] Liu, K., Shi, X., Kumar, A., Zhu, L., Natarajan, P. (2016). Temporal learning and sequence modeling for a job recommender system. ACM Press. <http://dx.doi.org/10.1145/2987538.2987540>
- [19] Hidasi, B., Karatzoglou, A. (2018). Recurrent neural networks with Top-k gains for session-based recommendations. Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18. ACM Press. <http://dx.doi.org/10.1145/3269206.3271761>
- [20] Smirnova, E., Vasile, F. (2017). Contextual sequence modeling for recommendation with recurrent neural networks. Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems - DLRS 2017. ACM Press. <http://dx.doi.org/10.1145/3125486.3125488>
- [21] Tang, S., Wu, Z., Chen, K. (2016). Movie recommendation via BLSTM. lecture notes in computer science. Springer International Publishing, 31: 269-279. [http://dx.doi.org/10.1007/978-3-319-51814-5\\_23](http://dx.doi.org/10.1007/978-3-319-51814-5_23)
- [22] Zhu, A., Meng, Y., Zhang, C. (2017). An improved Adam Algorithm using look-ahead. Proceedings of the 2017 International Conference on Deep Learning Technologies - ICDLT '17. ACM Press. <http://dx.doi.org/10.1145/3094243.3094249>
- [23] Reddy, K.V.R., Rao, S.B., Raju, K.P. (2018). Handwritten Hindi digits recognition using convolutional neural network with RMSprop optimization. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) IEEE. <http://dx.doi.org/10.1109/iccons.2018.8662969>
- [24] Coba, L., Symeonidis, P., Zanker, M. (2018). Replicating and improving Top-N recommendations in open source packages. Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics - WIMS '18. ACM Press. <http://dx.doi.org/10.1145/3227609.3227671>
- [25] Symeonidis, P., Zioupos, A. (2016). Matrix and tensor factorization techniques for recommender systems. Springer Briefs in Computer Science. Springer International Publishing. <http://dx.doi.org/10.1007/978-3-319-41357-0>
- [26] Lerche, L., Jannach, D. (2014). Using graded implicit feedback for Bayesian personalized ranking. Proceedings of the 8th ACM Conference on Recommender systems - RecSys'14. ACM Press. <http://dx.doi.org/10.1145/2645710.2645759>
- [27] Papagelis, M., Plexousakis, D. (2005). Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. Engineering Applications of Artificial Intelligence, 8(7): 781-789. <http://dx.doi.org/10.1016/j.engappai.2005.06.010>
- [28] Shani, G., Gunawardana, A. (2010). Evaluating recommendation systems. Recommender Systems Handbook. Springer, 5: 257-297. [http://dx.doi.org/10.1007/978-0-387-85820-3\\_8](http://dx.doi.org/10.1007/978-0-387-85820-3_8)