COMPUTATIONAL INTELLIGENCE AND MACHINE LEARNING
e-ISSN: 2582-7464

Computational Intelligence and Machine Learning
Vol-1, Issue-1, August 2020

Recurrent Neural Networks for Recommender Systems

Ankit Rath[1]*, Subhrajyoti Ranjan Sahu[2]

[1] B.Tech, Biotechnology, Gandhi Institute of Engineering and Technology, Gunupur, Odisha, India.
[2] B. Tech.,ECE, Nalanda Institute of Technology, Bhubaneswar, India.
*Corresponding Author Email: [1] rheaansh@gmail.com

Abstract

The Internet is becoming one of the biggest sources of information in recent years, keeping people updated about everyday events. The information available on it is also growing, with the increase in the use of the Internet. Due to this, it takes a great deal of time and effort to locate relavent knowledge that the user wants. Recommender systems are software mechanisms that automatically suggest relavent user-needed information. Recurrent Neural Networks has lately gained importance in the field of recommender systems, since they give improved results in building deep learning models with sequential data. Unlike conventional recommendendation models, RNN models more easily capture irregular and complex user-item relations. This paper provides a thorough analysis of the research content of recommendation systems based on RNN models.

Keywords

Gated Recurrent Unit, Long Short Term Memory, Recommendations, Recommender systems, Recurrent Neural Networks.

INTRODUCTION

The increase Internet usage contributes to large amount of online data that can be used by users to collect knowledge about interesting items. It is a hectic job to filter out relevant information from such voluminous data. Recommender systems (RS) helps in soliving the problem of information overload. The principle idea behind RS is to suggest things or items to the user, then receive a user-customized array of items that are relavent to them [1]. Recommender systems are very important in some industries because they provide a way to stand out from rivals significantly. Traditional recommender systems take user's previous interactions to predict their responses for new items suggests relevant items to the users. Traditional methods are classified into three types [2]: content-based recommendation, collaborative filtering-based recommendation, and hybrid recommendation methods.

Content based RS uses user history and item features inorder to make recommendations. Recommendations are made by finding items identical to those in user's past activities [6]. Collaborative filtering based RS methods (CF) make recommendations by evaluating existing users with the same preferences as the target user [4]. They are classified into: memory based approaches and model based approaches [5]. Memory based approaches take the complete historical data to make recommendations. Memory based CF approach is divided into user-based and item-based methods. These methods face the cold start problem where it cannot suggest new items and cannot make recommendations to a new user. Model based methods train a model using the user and item information and this model makes the recommendations. These models learn more complex relations and provides improved recommendations. Still these models are linear and fails to learn some relations. Hybrid recommender systems aim to combine the complementary strengths of content based and CF based approaches to build an overall more effective framework [7].

Deep learning methods are becoming one of the successful methods that are able to automatically extract and learn complex relations between users and items [1]. Multi-layered, nonlinear networks are capable of extracting complex features from a vast volume of data [3]. Deep learning has significantly redefined the recommendation systems and offers more ways to boost quality recommendations. Modern developments in deep learning-based recommender systems have gained considerable attention by overcoming limitations on conventional models and providing recommendations of high quality. Recurrent neural networks (RNN) is a type of deep learning model that give improved results in sequence modeling. These are considered to have memories and make decisions by adding user history inputs. This paper provide complete analysis on RNN in the area of recommender systems.

The rest of this paper is arranged as follows. Section 2 discusses background concepts. In section 3, covers surveys literature related to the subject. In section 4, we discuss some of the relavent projects in the field of recommender systems using RNN. In the end, section 5 concludes the paper.

BACKGROUND CONCEPTS

This section covers basic backkground concepts regarding Recurrent neural networks.

Deep Learning Techniques

Deep learning is a recent branch of machine learning research [8]. Deep neural networks (DNN) are multilayer neural networks with multiple hidden layers. For recommender systems there are many ways to use deep learning. Deep neural network models works like human brain in analyzing images, texts and sounds. DNNs are able to predict the next step based on past behaviour and content. They can be divided into supervised and unsupervised [9].

Table 1. Some of the deep learning techniques used in recommender systems

Approach	Details
Convolutional Neural Network (CNN)	It is a type of deep neural network used mainly in image processing. This uses convolution and pooling operations. Many recommendation models based on CNN use CNNs for extraction of the functionality.
Restricted Boltzmann Machine (RBM)	RBM consists of two layers, a visible layer and a hidden layer. Through learning the underlying intrinsic meaning of data they can solve complex learning problems.
Autoencoder (AE)	It is a neural network of three layers with the same number of nodes in both input and output layers. They are unsupervised networks which tries to reconstruct the input signal. Autoencoders in recommender systems helps to learn two dimensional feature representations.
Recurrent Neural Networks (RNN)	They are feed forward networks used in modeling sequential data.

Recurrent Neural Networks (RNNs)

RNNs introduce an intrestind spin to the traditional neural networks. RNNs are intended to take a set of inputs that do not have a fixed size limit. They can memorize portions of the given data and make predictions. They have implemenations in natural language processing and speech recognition [11].

RNNs remembers the past, and the decisions it makes based on those past memories. Standard feed forward networks "remember" things as well, however what they remember is what they learned during the training. The concept behind RNNs is to use sequential information; they have the ability to store information about what has been measured recently. This is Long Short Term Memory. RNN output is calculated by weights applied to inputs such as a regular neural network as well as a hidden state vector that defines the context based on the previous input or output. The same input will therefore generate a varying result depending on the previous inputs and outputs in the series. RNNs are called recurrent, since for each part of a sequence they perform the same operation, with the output based on the previous computations [1]. LSTM and GRU are common types of RNN. Standard machine learning models may be used for sequence-based recommendations, but in modelling longer-term relationships, GRU or LSTM recurrent neural networks tend to be most effective.

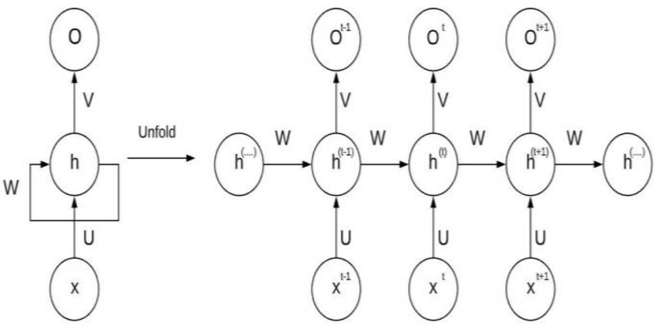

Fig. 1. Standard RNN and unfolded RNN

Recurrent Neural Networks have short-term memory. From the outset, RNN's can leave important information out. RNN's have the problem of vanishing gradients during back propagation. Gradients are the values used to change weights in a neural network. The major issue regarding the vanishing gradient is that as it propagates back through time, the gradient decreases. When the value of gradient is so small, it doesn't provide that much learning. So, layers that get a smaller gradient update cease learning in recurrent neural networks. Typically those are earlier layers. This implies that a recurrent network forgets the thing it learnt earlier in a longer sequence, and therefore has a short-term memory. Long Short Term Memory and Gated Recurrent Unit were developed as the remedy for the memory problem in recurrent networks. They resolve this problem by using gates, which can control information flow. The gates have the ability to decide which data in the sequence has to be retained or to be discarded so that important data is passed along the lengthy chain of sequences to make predictions. Both these models solve the problem of vanishing gradient by re-parameterizing the RNN.

Long Short-Term Memory (LSTM)

The basic concept of LSTM is the cell state and its numerous gates. A typical LSTM unit consists of a forget gate, an input gate and an output gate. During the training, the gates learn what information is necessary to keep or forget. With the input gate activation, the LSTM cell multiplies its input, and the forget gate multiplies the previous values.

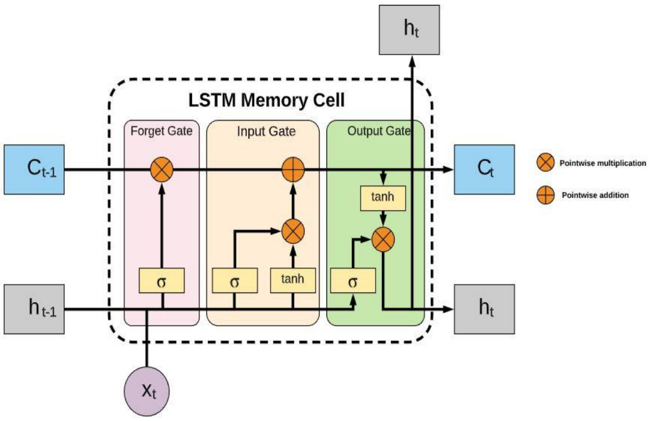

Fig. 2. Standard LSTM memory cell

Forget gate. This gate solves which information should be discarded or retained. The sigmoid function is given the preceding hidden state and present input information. The values range from zero to one, closer to zero mean it should be lost and closer to one suggest it should be kept.

Input gate. The cell status is changed via the input doors. Firstly, a sigmoid function is given the earlier hidden state and present information. This determines the values to be updated by converting the values in a range from zero to one. Zero is insignificant and one does mean important. In the tanh function, to help supervise the network, you also move the current input and the hidden state to compress values in the range (-1,1). The tanh function output is scaled up to the sigmoid function output. The knowledge that is to be held from the tanh function is calculated using the sigmoid function output.

Cell State. The state of the cell is assessed using the data being stored. First point wise multiplication is done between the cell state and the forget vector. Low cell state value is likely if compounded by values that are closer to zero. Now point wise addition performed on the input gate output, this updates the cell status. The cell state now includes new values essential for the network.

Output state. It decides the network's next, hidden state. Note that the hidden state consists of preceding input data and is used for predicting. Next, the hidden state preceding and present input are passed into the sigmoid function. The recently changed state of the cell is then transferred to the feature tanh. To decide which data will pass the hidden state, we multiply the tanh-function output with the sigmoid-function output. Therefore the output is the hidden state. The latest cell state and the hidden state are passed on to the corresponding phase in time.

The Forget gate decides which data should be kept from the earlier steps. From the present step, the input gate determines what data needs to be added. The next hidden state is decided by the gate to the output.

Gated Recurrent Unit (GRU)

The GRU, a newer RNN version, is nearly similar to the LSTM network. GRU has a simpler architecture than LSTM as it replaces the forget gate and the LSTM input gate with an update gate and a fusion of the hidden cell state. GRU has no cell state, and the hidden state is used for the transmission of data. The two gates in GRU are the reset gate and update gate. GRU works better on a variety of purposes than LSTM [10]. GRU has less parameter as compared to LSTM due to the fact that it lacks an output gate.

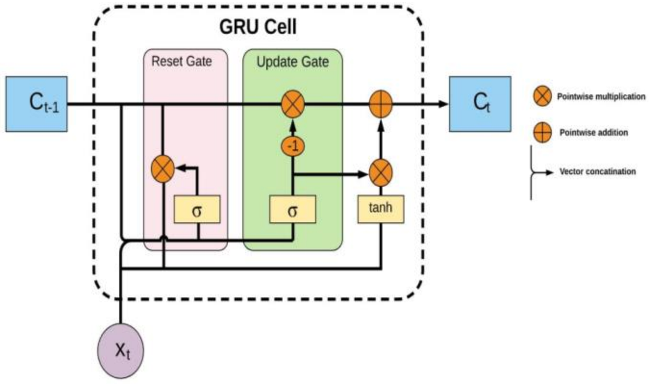

Fig. 3. Standard GRU cell

Update gate. It functions analogously to the forget gate and input gate of an LSTM. This determines which data should be discarded and what new ones should be added.

Reset gate. It is used to decide the amount of prior data that has to be kept.

GRU employs fewer memory and preparation criteria. Use longer series, LSTM is more reliable on the data collection. GRU's have fewer tensor operations so they're a little faster to train than LSTM's. Which one is better, no clear winner has been found. GRU employs fewer memory and preparation criteria. LSTM is more accurate to use the longer series on the dataset.

RNN IN RECOMMENDER SYSTEMS

RNNs are ideal for modelling data containing sequences. They are different from traditional neural networks because RNN has loops and memories that records preceding computations. Variants like LSTM and GRU are introduced as a solution to the issue of vanishing gradients. The underlying concept of recommender systems using RNNs is that they can model the effect of users past sequential activity on user recent actions, after which propose user-friendly things and forecast user behavior [12]. RNNs can model sequential data without notion of time periods between successive events, taking into consideration varying time intervals. Nevertheless, time deltas can be highly useful knowledge explicitly for recommendation tasks. For example , If two consumer items are remote, the first item may not be a fair predictor, because customer preferences have obviously evolved over time. RNNs can be developed to describe user actions more accurately and identify interconnections among consumption actions than those of existing recommender systems that are unable to reflect temporal dynamics in the interests of users.

POSSIBLE RESEARCH DIRECTIONS

In recent years , since RNN plays important role in modelling sequencial data, they are commonly used in the area of natural language procesing, session based recommender systems and so on. This session discusses some of the relavent projects in the field of recommender systems using RNN.

Learning Path Recommender System Using RNN

Recently, education in programming has gained more prominence due to the increasing demands for programming and IT skills. The lack of teaching materials and human capital has made it difficult to meet the increasing demand for programming instruction. A solution to the shortage of qualified teachers is by using machine learning methods to help the learner. AccordinglyUsing a recurrent neural network, a recommendation framework is introduced that proposes a learning route focused on the skill charts of the learner. In short, the learning path is created using a trial and error approach from the previous sessions of the learner and the ability charts of the learner are used to measure their existing knowledge. This is an approach to the development of a recommendation framework for the learning path through the use of ability charts and their implementation through a recurrent neural network based on a sequential model [13]. This also provides a theoretical assessment based on data from an e-learning program.

Sequential User Based Recommender Systems

A unique characteristic of recommender systems is considered the extent of how Recurrent Neural Networks can be applied to it. It provides a particular type of deep learning models with multiple properties that make them attractive for sequence modelling. Deep Learning automatically generates suitable data representations without manual feature engineering and identifies the sequential recommendation issue. Despite of the long-term emphasis on time-dependent models, overall sequence modeling in the recommender schemes was less discussed. Additionally, individual consumers can be represented in sequences of items consumed in a new type of Gated Recurrent Network to effectively produce personalized next item recommendations. The advantages of gated RNNs are modelled in terms of the periodic regularity of consumption sequences and an external feedback layer for individual users in a network [14]. Deep learning is used in traditional RS techniques during pre-processing steps. A recurrent user-based neural network is to incorporate user value deeply into the gating process.

Session Based Recommender System

In several popular web platforms (for example e-commerce, multimedia) and recommendation frameworks, session-based recommendations are highly important. Recently it has been shown that Recurrent Neural Networks performs really well in session-based environments. Even when user identifiers are tough to obtain through in most of the session-based recommendation models, there are also domains where user profiles are easily obtainable. The issue with customizing this type of RS is resolved by introducing a model-based Hierarchical RNN, which expands beyond RNN-based session modeling with an additional GRU stage that models user behavior through sessions and the progression of their preferences across time [15]. HRNNs offer a streamlined way to move the information collected from the long-term complexities of the user's actions to the session level and thereby offer tailored session-based recommendations to the user's return. The HRNNs model outperforms all session-based RNNs and other simple session-based suggestion optimization approaches on real-world datasets of various character.

Recommender Systems using Sentiment Analysis

The recommender systems play a significant role in delivering product recommendations by leveraging the publicly accessible item details across online social networks. Automatic recommender framework on the cloud will recommend products by providing feedback across the web portal depending on user queries. Recommenders programs can recommend items by offering feeling information about the comments that people have written about the objects across various social media sites. For a given paragraph, features can be extracted in recurrent neural networks and these features are interpreted in a high-dimensional vector space, and then applied to the neural network. Recurrent neural networks possess the ability to effectively understand the sentence structure. This neural network architecture is also ideally suited for study of emotions [16].

CONCLUSION

The drastic growth in the quantity of data generated online necessitates the need for clever technologies and software that can efficiently and productively store, handle, view and interpret information for optimal gain for consumers. Recurrent neural network-based recommender systems are suitable tools that can help the information-seeking process easy. Comparison to standard recommender systems, RNN-based recommender systems can use deep learning techniques to automatically learn user and item feature vector by combining multiple forms of diverse multi-source data, modeling user behaviour sequence patterns, more accurately representing the diverse user tastes and increasing recommendation accuracy.

REFERENCES

[1] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," in IEEE Access, vol. 6, pp. 69009-69022, 2018, doi: 10.1109/ACCESS.2018.2880197.

[2] M. Morisio, "Hybrid recommender systems: A systematic literature review", Intell. Data Anal., vol. 21, no. 6, pp. 1487-1524, 2017.

[3] Y.-X. Peng et al., "Cross-media analysis and reasoning: Advances and directions", Frontiers Inf. Technol., vol. 18, no.

5, pp. 44-57, 2017.
[4] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, “Collaborative Filtering Recommender Systems”, Foundations and Trends in Human–Computer Interaction Vol. 4, No. 2 (2010) 81–173.
[5] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” New Page, vol. 7, no. 7, pp. 43–52, 2013.
[6] J. Ben Schafer, J. Konstan and J. Riedl, "Recommender systems in e-commerce", Proc. 1st ACM Conf. Electron. Commerce, pp. 158-166, 1999.
[7] P. Kosmides, C. Remoundou, I. Loumiotis, E. Adamopoulou, M. Theologou and K. Demestichas, "A location recommender system for location-based social networks", Proc. IEEE Int. Conf. Math. Comput. Sci. Ind., pp. 277-280, Sep. 2015.
[8] J. Zhu, Y. Shan, J. C. Mao, D. Yu, H. Rahmanian and Y. Zhang, Deep embedding forest: Forest-based serving with deep embedding features, Mar. 2017
[9] G. Zhou et al., Deep interest network for click-through rate prediction, Jun. 2017
[10] David Zhan Lin and Gurbir Singh , “A Recurrent Neural Network Based Recommendation System”
[11] Robin Devooght and Hugues Bersini, “Collaborative Filtering with Recurrent Neural Networks”, arXiv, 3rd Jan 2017
[12] C. Wu, J. Wang, W. Liu and J. Liu, "Recurrent neural network based recommendation for time heterogeneous feedback", Knowl.-Based Syst., vol. 109, pp. 90-103, Oct. 2016.
[13] Tomohiro Saito and Yutaka Watanobe, “Learning Path Recommender System based on Recurrent Neural Network”, 9th International conference on Awareness Science and Technology (iCAST), 2015
[14] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In Pro- ceedings of RecSys ’17, Como, Italy, August 27-31, 2017, 9 pages.
[15] Quadrana M, Karatzoglou A, Hidasi B, Cremonesi P. Personalizing session-based recommendations with hierarchical recurrent neural networks. InProceedings of the Eleventh ACM Conference on Recommender Systems 2017 Aug 27 (pp. 130-137).
[16] G. Preethi, P.V. Krishna,S. Obaidat Mohammad, V.Saritha and Y. Sumanth, “ Application of Deep Learning to Sentiment Analysis for recommender system on cloud”, International Conference on Computer, Information and Telecommunication Systems (CITS) , 2017, pp. 93-97.