



NTNU – Trondheim
Norwegian University of
Science and Technology

Recurrent Neural Network for Session-based Recommendations

Ole Steinar Lillestøl Skrede

December 2016

Department of Computer and Information Science
Norwegian University of Science and Technology

Supervisor: Massimiliano Ruocco

Abstract

Contents

List of Figures	1
1 Introduction	2
1.1 Motivation	2
1.2 Objectives	4
1.3 Background	4
1.3.1 Recommender systems	4
1.3.2 Collaborative filtering	6
1.3.3 Session-based recommender systems	6
2 State of the Art	8
3 Core	9
4 Experiments and results	10
4.1 Experiments	10
4.2 Results	10
4.3 Discussion	10
5 Conclusion and further work	11
5.1 Conclusion	11
5.2 Further work	11
A List of Acronyms	12
Bibliography	13

List of Figures

1.1	Recommendations from Amazon (left) and YouTube (right) when looking at Xbox One.	5
1.2	Illustration of collaborative filtering	7

Chapter 1

Introduction

There are vast amount of information and products available on the web. Even within a single websites the number of items can be overwhelming for users. Examples of sites where this applies are news sites, streaming services, and e-commerce sites. Recommender systems can help creating a better user experience by helping users find what they are looking for and are interested in. They can also help businesses in different ways, like showing targeted ads or help offering a better product.

In this paper, we will explore recommender systems, specifically session-based recommender systems. We will look at the use of deep learning and recurrent neural networks(RNN) for this, as compared to classical methods.

This introduction chapter introduces session-based recommender systems, some of the challenges in the domain, recurrent neural networks and how they fit into the problem. The second chapter, State of the art, looks at different methods that exist, but focuses on what has been done in terms of deep learning and RNNs. In the third chapter, Core, we explain our own work. Our experiments and results are laid out in the fourth chapter, Experiments and results. Lastly follows a the conclusion and further work, which summarize what has been done, which challenges remain and needs to be studied further.

1.1 Motivation

As mentioned, recommender systems can help both users looking for content and the providers of the content. Users are generally interested in finding what they are looking for as easy as possible, or to be shown products or content that interests them, but which they normally would not have discovered on their own. Spotify helps users discover new content introducing them to new music, tailored to the

user, through their Discover Weekly[1] playlists. When a user buys something on an e-commerce site like Amazon, the site displays items to the user, that other users bought together with the original item. These are examples where recommender systems help users discover new content and help users find products they are interested in.

For the provider, or seller, there may be different reasons to use recommender systems depending on their business model. Google earns money when users click on ads, so it is important for them to show relevant ads with high probability of getting clicked by the user. Similarly, YouTube earns money by showing commercials in their videos, and thus they can earn more money by making the user watch more videos. This can be accomplished by suggesting videos that the user wants to watch. Or like in the Amazon example, suggesting additional items might make the buyer buy more than he had initially intended. Recommender systems are also used by Facebook and news feeds which try to filter out what is shown to the user.

In the examples above, recommendations are made based on the user profile, which has information about the user's interests. This profile is built up over time, as the user interacts with a system/site. Or in the Amazon example, the recommendation may only be based on a single or few items that the user bought. To make good recommendations in these scenarios, it is assumed that one has access to a user profile, but this might not always be the case. Multiple users may share an account, maybe the user is new, or maybe the user is not logged in at all. Sites can use cookies to get some historical data about the user when he is not logged in, but this is not very reliable either. There might not exist any cookies because the user is completely new to the site, or he may have deleted them. And even though we have access to cookies with information, the traffic might come from a computer shared by family members for example. Son and mom are probably not buying the same clothes, or listening to the same music. But even when we assume that it is possible to know who the user is, and what his interests are, he might be searching for content/products that is not related to his interests at all. An example of this could be a professional angler that wants to buy beginner equipment for his nephew, or maybe the angler suddenly has decided to start playing football and is looking for football equipment. Another example is a programmer who wants to listen to calm classical music while he works, but when he goes to the gym after work, he wants to listen to energetic techno. In these situations the recommender system will not be able to make good recommendations based on what it knows about the user. Only the current session might be relevant to what the user is looking for.

Recurrent neural networks intuitively fits well as a recommender system, especially in the session-based scenario. They do not suffer from the cold start scenario, they can perform well with minimal information about items, and they make it easier to extract good features [2]. Also, RNNs does not suffer from some of the assumptions that classical neural networks do. With recent advances in architectures for RNN, ways to circumvent training problems, and better hardware, RNNs have become very feasible to use. Much research has been done on RNNs recently, some of this research focuses on using RNN as a session-based recommender system. The results are promising, and it seems like RNNs are fully capable of competing with state of the art recommender systems.

1.2 Objectives

The goal of this paper and the work described in it, is to implement an RNN and test how it performs as a recommender system. We explore recommender systems, particularly those that deal with session-based recommendation. Specifically we look at the use of RNNs as session-based recommender systems. We want to know if they can compete with other models currently in use. Furthermore, we look at different architectures and ways of optimizing them to improve results. Our work is inspired, mainly, by the work in [3] and [4]. The first paper explores the use of an RNN for session-based recommendations compared to other methods. The second paper explores how adding context information as input to the network can improve the results.

1.3 Background

In this section we explain, on a high level, terms and concepts used in this paper.

1.3.1 Recommender systems

As touched upon in 1.1, recommender systems are systems that try to predict a users evaluation of an item, or what items a user is interested in interacting with. They can be, and are, used in many areas. Some examples are music, movies, news, restaurants, recipes, online shopping, and dating. Figure 1.1 shows an example of recommendations made by Amazon and YouTube when looking at a Xbox One.

Note that there is a subtle difference between predicting which items a user will like, and which items he is interested in interacting with. The former case involves predicting what rating a user will give a movie or whether a user is likely

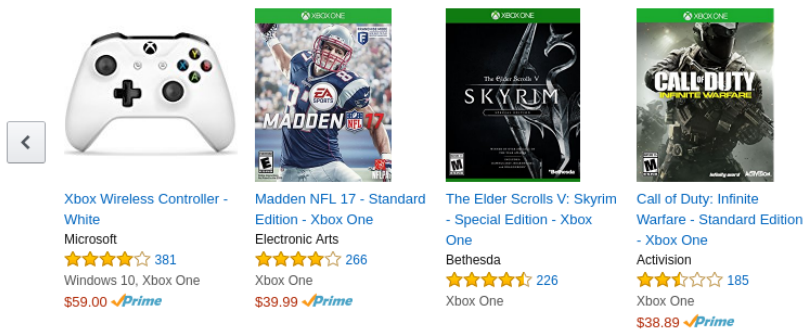
Frequently Bought Together



i These items are shipped from and sold by different sellers. [Show details](#)

- ☒ **This item:** Xbox One S 500GB Console - Battlefield 1 Bundle by Microsoft Xbox One **\$289.95**
- ☒ **Madden NFL 17 - Standard Edition - Xbox One** by Electronic Arts Xbox One **\$39.99**
- ☒ **Xbox Wireless Controller - White** by Microsoft Windows 10, Xbox One **\$59.00**

Customers Who Bought This Item Also Bought



Unboxing & Setup: Microsoft Xbox One
Jack Stavris
480,388 views



Xbox One vs PS4 - Full Comparison
TechSmartt
5,540,108 views



Xbox One S Review
IGN
279,069 views



Microsoft Xbox One unboxing, setup &
Gamebits
2,190,897 views

Figure 1.1: Recommendations from Amazon (left) and YouTube (right) when looking at Xbox One.

to purchase one or any item. On the other hand, the latter case does not worry about how the user will interact with an item, but whether the user will choose to interact with the item, given the choice. To exemplify this, let's look at a movie streaming site. Should the site recommend movies that the user will rate highly if he watches them, or should it recommend movies that the user is likely to watch independent of how he will rate it? The answer will probably depend on the business and other factors, the point is that there is a difference between the two approaches. In the movie example, it is reasonable to think that how a user would rate a movie is less dynamic than what he wants to watch. His ratings would probably depend a lot on his personal preferences, which usually change slowly over time. What he is interested in watching at any point, however, might depend more on circumstances. This was illustrated earlier with the worker who wanted to listen to different genres at during work and his workout session.

It is therefore clear that the goal of recommender systems may vary with different businesses and settings. In this paper we are mainly concerned with predicting the next item the user will choose to interact with.

There are two classical approaches to recommender systems, collaborative filtering and content-based filtering. These two methods can be combined into a hybrid approach.

1.3.2 Collaborative filtering

Collaborative filtering uses information about users' preferences to recommend items highly rated by similar users. To illustrate, let us look at a movie recommendation system. A user logs into a website where he can rate movies he has seen and the site recommends movies to the user based on his ratings. With the collaborative filtering approach, the user needs to rate some movies to get good recommendations, and as the user rates more and more movies he has seen, the system is able to make better recommendations. To make the recommendations, the system groups together similar users. A user is then recommended movies that he has not seen and that were rated highly by other similar users. The system decides whether users are similar by looking at how like-minded they are, that is, how similar they rate movies. This is illustrated in figure 1.2.

1.3.3 Session-based recommender systems

Session-based recommender systems try to do predictions mostly based on the current user session. Sessions are often limited in time, and, as we have seen, the user's interests might vary between sessions. Generally we have no information about


























				
				
				
				
				
				

Figure 1.2: Illustration of collaborative filtering

Source: https://en.wikipedia.org/wiki/Collaborative_filtering#/media/File:Collaborative_filtering.gif

the user, and since the sessions are short, we do not get much information either. There may be many reasons for this lack of information. Some examples are new users, users sharing accounts, and small e-commerce sites where each unique user only visits it a couple of times. Furthermore, even though we have access to information about the users interest, this might be of minimal use if the users interests vary greatly between sessions. A session usually consists of multiple user actions on a number of items, in the order the actions happen, constrained to a short timespan. What "short timespan" means will depend on the domain, but is often on the scale of minutes.

Chapter 2

State of the Art

Chapter 3

Core

Chapter 4

Experiments and results

4.1 Experiments

4.2 Results

4.3 Discussion

Chapter 5

Conclusion and further work

5.1 Conclusion

5.2 Further work

Appendix A

List of Acronyms

RNN Recurrent Neural Network

RBM Restricted Boltzmann machines

LSTM Long Short Term Memory

FFNN Feed Forward Neural Networks

Bibliography

- [1] Spotify. Introducing discover weekly: your ultimate personalised playlist. <https://press.spotify.com/it/2015/07/20/introducing-discover-weekly-your-ultimate-personalised-playlist/>.
- [2] Tobias Lang. Deep learning for understanding consumer histories. <https://tech.zalando.com/blog/deep-learning-for-understanding-consumer-histories/>.
- [3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *CoRR*, abs/1511.06939, 2015.
- [4] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. Context-aware sequential recommendation. *CoRR*, abs/1609.05787, 2016.