Deep Learning based Recommender System: A Survey and New Perspectives

SHUAI ZHANG, University of New South Wales
LINA YAO, University of New South Wales
AIXIN SUN, Nanyang Technological University
YI TAY, Nanyang Technological University

With the ever-growing volume of online information, recommender systems have been an effective strategy to overcome such information overload. The utility of recommender systems cannot be overstated, given its widespread adoption in many web applications, along with its potential impact to ameliorate many problems related to over-choice. In recent years, deep learning has garnered considerable interest in many research fields such as computer vision and natural language processing, owing not only to stellar performance but also the attractive property of learning feature representations from scratch. The influence of deep learning is also pervasive, recently demonstrating its effectiveness when applied to information retrieval and recommender systems research. Evidently, the field of deep learning in recommender system is flourishing. This article aims to provide a comprehensive review of recent research efforts on deep learning based recommender systems. More concretely, we provide and devise a taxonomy of deep learning based recommendation models, along with providing a comprehensive summary of the state-of-the-art. Finally, we expand on current trends and provide new perspectives pertaining to this new exciting development of the field.

CCS Concepts: •**Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Recommender System; Deep Learning; Survey

ACM Reference format:
Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2018. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 1, 1, Article 1 (July 2018), 35 pages.
DOI: 0000001.0000001

1 INTRODUCTION

Recommender systems are an intuitive line of defense against consumer over-choice. Given the explosive growth of information available on the web, users are often greeted with more than countless products, movies or restaurants. As such, personalization is an essential strategy for facilitating a better user experience. All in all, these systems have been playing a vital and indispensable role in various information access systems to boost business and facilitate decision-making process [69, 121] and are pervasive across numerous web domains such as e-commerce and/or media websites.

In general, recommendation lists are generated based on user preferences, item features, user-item past interactions and some other additional information such as temporal (e.g., sequence-aware recommender) and

Yi Tay is added as an author later to help revise the paper for the major revision.

Author's addresses: S. Zhang and L. Yao, University of New South Wales; emails: shuai.zhang@unsw.edu.au; lina.yao@unsw.edu.au; A. Sun and Y. Tay, Nanyang Technological University; email: axsun@ntu.edu.sg; ytay017@e.ntu.edu.sg;

DOI: 0000001.0000001

spatial (e.g., POI recommender) data. Recommendation models are mainly categorized into collaborative filtering, content-based recommender system and hybrid recommender system based on the types of input data [1].

Deep learning enjoys a massive hype at the moment. The past few decades have witnessed the tremendous success of the deep learning (DL) in many application domains such as computer vision and speech recognition. The academia and industry have been in a race to apply deep learning to a wider range of applications due to its capability in solving many complex tasks while providing start-of-the-art results [27]. Recently, deep learning has been revolutionizing the recommendation architectures dramatically and brings more opportunities to improve the performance of recommender. Recent advances in deep learning based recommender systems have gained significant attention by overcoming obstacles of conventional models and achieving high recommendation quality. Deep learning is able to effectively capture the non-linear and non-trivial user-item relationships, and enable the codification of more complex abstractions as data representations in the higher layers. Furthermore, it catches the intricate relationships within the data itself, from abundant accessible data sources such as contextual, textual and visual information.

Pervasiveness and ubiquity of deep learning in recommender systems. In industry, recommender systems are critical tools to enhance user experience and promote sales/services for many online websites and mobile applications [20, 27, 30, 43, 113]. For example, 80 percent of movies watched on Netflix came from recommendations [43], 60 percent of video clicks came from home page recommendation in YouTube [30]. Recently, many companies employ deep learning for further enhancing their recommendation quality [20, 27, 113]. Covington et al. [27] presented a deep neural network based recommendation algorithm for video recommendation on YouTube. Cheng et al. [20] proposed an App recommender system for Google Play with a wide & deep model. Shumpei et al. [113] presented a RNN based news recommender system for Yahoo News. All of these models have stood the online testing and shown significant improvement over traditional models. Thus, we can see that deep learning has driven a remarkable revolution in industrial recommender applications.

The number of research publications on deep learning based recommendation methods has increased exponentially in these years, providing strong evidence of the inevitable pervasiveness of deep learning in recommender system research. The leading international conference on recommender system, RecSys[1], started to organize regular workshop on deep learning for recommender system[2] since the year 2016. This workshop aims to promote research and encourage applications of deep learning based recommender system.

The success of deep learning for recommendation both in academia and in industry requires a comprehensive review and summary for successive researchers and practitioners to better understand the strength and weakness, and application scenarios of these models.

What are the differences between this survey and former ones? Plenty of research has been done in the field of deep learning based recommendation. However, to the best of our knowledge, there are very few systematic reviews which well shape this area and position existing works and current progresses. Although some works have explored the recommender applications built on deep learning techniques and have attempted to formalize this research field, few has sought to provide an in-depth summary of current efforts or detail the open problems present in the area. This survey seeks to provide such a comprehensive summary of current research on deep learning based recommender systems, to identify open problems currently limiting real-world implementations and to point out future directions along this dimension.

In the last few years, a number of surveys in traditional recommender systems have been presented. For example, Su et al. [138] presented a systematic review on collaborative filtering techniques; Burke et al. [8] proposed a comprehensive survey on hybrid recommender system; Fernández-Tobías et al. [40] and Khan et al. [74] reviewed the cross-domain recommendation models; to name a few. However, there is a lack of extensive

[1] https://recsys.acm.org/
[2] http://dlrs-workshop.org/

review on deep learning based recommender system. To the extent of our knowledge, only two related short surveys [7, 97] are formally published. Betru et al. [7] introduced three deep learning based recommendation models [123, 153, 159], although these three works are influential in this research area, this survey lost sight of other emerging high quality works. Liu et al. [97] reviewed 13 papers on deep learning for recommendation, and proposed to classify these models based on the form of inputs (approaches using content information and approaches without content information) and outputs (rating and ranking). However, with the constant advent of novel research works, this classification framework is no longer suitable and a new inclusive framework is required for better understanding of this research field. Given the rising popularity and potential of deep learning applied in recommender system, a systematic survey will be of high scientific and practical values. We analyzed these works from different perspectives and presented some new insights toward this area. To this end, over 100 articles were shortlisted and classified in this survey.

How do we collect the papers? In this survey, we collected over a hundred of related papers. We used Google Scholar as the main search engine, we also adopted the database, Web of Science, as an important tool to discover related papers. In addition, we screened most of the related high-profile conferences such as NIPS, ICML, ICLR, KDD, WWW, SIGIR, WSDM, RecSys, etc., just to name a few, to find out the recent work. The major keywords we used including: recommender system, recommendation, deep learning, neural networks, collaborative filtering, matrix factorization, etc.

Contributions of this survey. The goal of this survey is to thoroughly review literature on the advances of deep learning based recommender system. It provides a panorama with which readers can quickly understand and step into the field of deep learning based recommendation. This survey lays the foundations to foster innovations in the area of recommender system and tap into the richness of this research area. This survey serves the researchers, practitioners, and educators who are interested in recommender system, with the hope that they will have a rough guideline when it comes to choosing the deep neural networks to solve recommendation tasks at hand. To summarize, the key contributions of this survey are three-folds: (1) We conduct a systematic review for recommendation models based on deep learning techniques and propose a classification scheme to position and organize the current work; (2) We provide an overview and summary for the state-of-the-arts. (3) We discuss the challenges and open issues, and identify the new trends and future directions in this research field to share the vision and expand the horizons of deep learning based recommender system research.

The remaining of this article is organized as follows: Section 2 introduces the preliminaries for recommender systems and deep neural networks, we also discuss the advantages and disadvantages of deep neural network based recommendation models. Section 3 firstly presents our classification framework and then gives detailed introduction to the state-of-the-art. Section 4 discusses the challenges and prominent open research issues. Section 5 concludes the paper.

2 OVERVIEW OF RECOMMENDER SYSTEMS AND DEEP LEARNING

Before we dive into the details of this survey, we start with an introduction to the basic terminology and concepts regarding recommender system and deep learning techniques. We also discuss the reasons and motivations of introducing deep neural networks to recommender systems.

2.1 Recommender Systems

Recommender systems estimate users' preference on items and recommend items that users might like to them proactively [1, 121]. Recommendation models are usually classified into three categories [1, 69]: collaborative filtering, content based and hybrid recommender system. Collaborative filtering makes recommendations by learning from user-item historical interactions, either explicit (e.g. user's previous ratings) or implicit feedback (e.g. browsing history). Content-based recommendation is based primarily on comparisons across items' and users' auxiliary information. A diverse range of auxiliary information such as texts, images and videos can be taken into account. Hybrid model refers to recommender system that integrates two or more types of recommendation strategies [8, 69].

Suppose we have M users and N items, and R denotes the interaction matrix and $\hat{R}$ denotes the predicted interaction matrix. Let r_{ui} denote the preference of user u to item i, and $\hat{r}_{ui}$ denote the predicted score. Meanwhile, we use a partially observed vector (rows of R) $\mathbf{r}^{(u)} = \{r^{u1}, ..., r^{uN}\}$ to represent each user u, and partially observed vector (columns of R) $\mathbf{r}^{(i)} = \{r^{1i}, ..., r^{Mi}\}$ to represent each item i. $\mathcal{O}$ and $\mathcal{O}^{-}$ denote the observed and unobserved interaction set. we use $U \in \mathcal{R}^{M \times k}$ and $V \in \mathcal{R}^{N \times k}$ to denote user and item latent factor. k is the dimension of latent factors. In addition, sequence information such as timestamp can also be considered to make sequence-aware recommendations. Other notations and denotations will be introduced in corresponding sections.

2.2 Deep Learning Techniques

Deep learning can be generally considered to be sub-field of machine learning. The typical defining essence of deep learning is that it learns *deep representations*, i.e., learning multiple levels of representations and abstractions from data. For practical reasons, we consider any neural differentiable architecture as *'deep learning'* as long as it optimizes a differentiable objective function using a variant of stochastic gradient descent (SGD). Neural architectures have demonstrated tremendous success in both supervised and unsupervised learning tasks [31]. In this subsection, we clarify a diverse array of architectural paradigms that are closely related to this survey.

- Multilayer Perceptron (MLP) is a feed-forward neural network with multiple (one or more) hidden layers between the input layer and output layer. Here, the perceptron can employ arbitrary activation function and does not necessarily represent strictly binary classifier. MLPs can be intrepreted as stacked layers of nonlinear transformations, learning hierarchical feature representations. MLPs are also known to be universal approximators.
- Autoencoder (AE) is an unsupervised model attempting to reconstruct its input data in the output layer. In general, the bottleneck layer (the middle-most layer) is used as a salient feature representation of the input data. There are many variants of autoencoders such as denoising autoencoder, marginalized denoising autoencoder, sparse autoencoder, contractive autoencoder and variational autoencoder (VAE) [15, 45].
- Convolutional Neural Network (CNN) [45] is a special kind of feedforward neural network with convolution layers and pooling operations. It can capture the global and local features and significantly enhancing the efficiency and accuracy. It performs well in processing data with grid-like topology.
- Recurrent Neural Network (RNN) [45] is suitable for modelling sequential data. Unlike feedforward neural network, there are loops and memories in RNN to remember former computations. Variants such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) network are often deployed in practice to overcome the vanishing gradient problem.
- Restricted Boltzmann Machine (RBM) is a two layer neural network consisting of a visible layer and a hidden layer. It can be easily stacked to a deep net. *Restricted* here means that there are no intra-layer communications in visible layer or hidden layer.
- Neural Autoregressive Distribution Estimation (NADE) [81, 152] is an unsupervised neural network built atop autoregressive model and feedforward neural networks. It is a tractable and efficient estimator for modelling data distribution and densities.
- Adversarial Networks (AN) [46] is a generative neural network which consists of a discriminator and a generator. The two neural networks are trained simultaneously by competing with each other in a minimax game framework.
- Attentional Models (AM) are differentiable neural architectures that operate based on soft content addressing over an input sequence (or image). Attention mechanism is typically ubiquitous and was

incepted in Computer Vision and Natural Language Processing domains. However, it has also been an emerging trend in deep recommender system research.

- Deep Reinforcement Learning (DRL) [106]. Reinforcement learning operates on a trial-and-error paradigm. The whole framework mainly consists of the following components: agents, environments, states, actions and rewards. The combination between deep neural networks and reinforcement learning formulate DRL which have achieved human-level performance across multiple domains such as games and self-driving cars. Deep neural networks enable the agent to get knowledge from raw data and derive efficient representations without handcrafted features and domain heuristics.

Note that there are numerous advanced model emerging each year, here we only briefly listed some important ones. Readers who are interested in the details or more advanced models are referred to [45].

2.3 Why Deep Neural Networks for Recommendation?

Before diving into the details of recent advances, it is beneficial to understand the reasons of applying deep learning techniques to recommender systems. It is evident that numerous deep recommender systems have been proposed in a short span of several years. The field is indeed bustling with innovation. At this point, it would be easy to question the *need* for so many different architectures and/or possibly even the utility of neural networks for the problem domain. Along the same tangent, it would be apt to provide a clear rationale of why each proposed architecture and to which scenario it would be most beneficial for. All in all, this question is highly relevant to the issue of task, domains and recommender scenarios. One of the most attractive properties of neural architectures is that they are (1) end-to-end differentiable and (2) provide suitable *inductive biases* catered to the input data type. As such, if there is an inherent structure that the model can exploit, then deep neural networks ought to be useful. For instance, CNNs and RNNs have long exploited the instrinsic structure in vision (and/or human language). Similarly, the sequential structure of session or click-logs are highly suitable for the inductive biases provided by recurrent/convolutional models [56, 143, 175].

Moreover, deep neural networks are also composite in the sense that multiple neural building blocks can be composed into a single (gigantic) differentiable function and trained end-to-end. The key advantage here is when dealing with *content-based* recommendation. This is inevitable when modeling users/items on the web, where multimodal data is commonplace. For instance, when dealing with textual data (reviews [202], tweets [44] etc.), image data (social posts, product images), CNNs/RNNs become indispensable neural building blocks. Here, the traditional alternative (designing modality-specific features etc.) becomes significantly less attractive and consequently the recommender system cannot take advantage of joint (end-to-end) representation learning. In some sense, developments in the field of recommender systems are also tightly coupled with advances research in related modalities (such as vision or language communities). For example, to process reviews, one would have to perform costly preprocessing (e.g., keyphrase extraction, topic modeling etc.) whilst newer deep learning-based approaches are able to ingest all textual information end-to-end [202]. All in all, the capabilities of deep learning in this aspect can be regarded as paradigm-shifting and the ability to represent images, text and interactions in a unified joint framework [197] is not possible without these recent advances.

Pertaining to the interaction-only setting (i.e., matrix completion or collaborative ranking problem), the key idea here is that deep neural networks are justified when there is a huge amount of complexity or when there is a large number of training instances. In [53], the authors used a MLP to approximate the interaction function and showed reasonable performance gains over traditional methods such as MF. While these neural models perform better, we also note that standard machine learning models such as BPR, MF and CML are known to perform reasonably well when trained with momentum-based gradient descent on interaction-only data [145]. However, we can also consider these models to be also neural architectures as well, since they take advantage of recent deep learning advances such as Adam, Dropout or Batch Normalization [53, 195]. It is also easy to see that, traditional recommender algorithms (matrix factorization, factorization machines, etc.) can also be expressed as neural/differentiable architectures [53, 54] and trained efficiently with a framework such as Tensorflow or Pytorch, enabling efficient GPU-emabled training and free automatic differentiation. Hence, in today's research climate (and even industrial), there is completely *no reason* to not used deep learning based tools for development of any recommender system.

To recapitulate, we summarize the strengths of deep learning based recommendation models that readers might bear in mind when try to employ them for practice use.

- **Nonlinear Transformation**. Contrary to linear models, deep neural networks is capable of modelling the non-linearity in data with nonlinear activations such as relu, sigmoid, tanh, etc. This property makes it possible to capture the complex and intricate user item interaction patterns. Conventional methods such as matrix factorization, factorization machine, sparse linear model are essentially linear models. For example, matrix factorization models the user-item interaction by linearly combining user and item latent factors [53]; Factorization machine is a member of multivariate linear family [54]; Obviously, SLIM is a linear regression model with sparsity constraints. The linear assumption, acting as the basis of many traditional recommenders, is oversimplified and will greatly limit their modelling expressiveness. It is well-established that neural networks are able to approximate any continuous function with an arbitrary precision by varying the activation choices and combinations [58, 59]. This property makes it possible to deal with complex interaction patterns and precisely reflect user's preference.
- **Representation Learning**. Deep neural networks is efficacious in learning the underlying explanatory factors and useful representations from input data. In general, a large amount of descriptive information about items and users is available in real-world applications. Making use of this information provides a way to advance our understanding of items and users, thus, resulting in a better recommender. As such, it is a natural choice to apply deep neural networks to representation learning in recommendation models. The advantages of using deep neural networks to assist representation learning are in two-folds: (1) it reduces the efforts in hand-craft feature design. Feature engineering is a labor intensive work, deep neural networks enable automatically feature learning from raw data in unsupervised or supervised approach; (2) it enables recommendation models to include heterogeneous content information such as text, images, audio and even video. Deep learning networks have made breakthroughs in multimedia data processing and shown potentials in representations learning from various sources.
- **Sequence Modelling**. Deep neural networks have shown promising results on a number of sequential modelling tasks such as machine translation, natural language understanding, speech recognition, chatbots, and many others. RNN and CNN play critical roles in these tasks. RNN achives this with internal memory states while CNN achieves this with filters sliding along with time. Both of them are widely applicable and flexible in mining sequential structure in data. Modelling sequential signals is an important topic for mining the temporal dynamics of user behaviour and item evolution. For example, next-item/basket prediction and session based recommendation are typical applications. As such, deep neural networks become a perfect fit for this sequential pattern mining task. This
- **Flexibility**. Deep learning techniques possess high flexibility, especially with the advent of many popular deep learning frameworks such as Tensorflow[3], Keras[4], Caffe[5], MXnet[6], DeepLearning4j[7], PyTorch[8], Theano[9], etc. Most of these tools are developed in a modular way and have active community and

[3] https://www.tensorflow.org/
[4] https://keras.io/
[5] http://caffe.berkeleyvision.org/
[6] https://mxnet.apache.org/
[7] https://deeplearning4j.org/
[8] https://pytorch.org/
[9] http://deeplearning.net/software/theano/

professional support. The good modularization makes development and engineering a lot more efficient. For example, it is easy to combine different neural structures to formulate powerful hybrid models, or replace one module with others. Thus, we could easily build hybrid and composite recommendation models to simultaneously capture different characteristics and factors.

2.4 On Potential Limitations

Are there really any drawbacks and limitations with using deep learning for recommendation? In this section, we aim to tackle several commonly cited arguments against the usage of deep learning for recommender systems research.

- **Interpretability.** Despite its success, deep learning is well-known to behave as black boxes, and providing explainable predictions seem to be a really challenging task. A common argument against deep neural networks is that the hidden weights and activations are generally non-interpretable, limiting explainability. However, this concern has generally been eased with the advent of neural attention models and have paved the world for deep neural models that enjoy improved interpretability [126, 146, 178]. While interpreting individual neurons still pose a challenge for neural models (not only in recommender systems), present state-of-the-art models are already capable of some extent of interpretability, enabling explainable recommendation. We discuss this issue in more detail in the open issues section.
- **Data Requirement.** A second possible limitation is that deep learning is known to be data-hungry, in the sense that it requires sufficient data in order to fully support its rich parameterization. However, as compared with other domains (such as language or vision) in which labeled data is scarce, it is relatively easy to garner a significant amount of data within the context of recommender systems research. Million/billion scale datasets are commonplace not only in industry but also released as academic datasets.
- **Extensive Hyperparameter Tuning.** A third well-established argument against deep learning is the need for extensive hyperparameter tuning. However, we note that hyperparameter tuning is not an exclusive problem of deep learning but machine learning in general (e.g., regularization factors and learning rate similarly have to be tuned for traditional matrix factorization etc) Granted, deep learning may introduce additional hyperparameters in some cases. For example, a recent work [145], attentive extension of the traditional metric learning algorithm [60] only introduces a single hyperparameter.

3 DEEP LEARNING BASED RECOMMENDATION: STATE-OF-THE-ART

In this section, we we firstly introduce the categories of deep learning based recommendation models and then highlight state-of-the-art research prototypes, aiming to identify the most notable and promising advancement in recent years.

3.1 Categories of deep learning based recommendation models

To provide a bird-eye's view of this field, we classify the existing models based the types of employed deep learning techniques. We further divide deep learning based recommendation models into the following two categories. Figure 1 summarizes the classification scheme.

- *Recommendation with Neural Building Blocks.* In this category, models are divided into eight subcategories in conformity with the aforementioned eight deep learning models: MLP, AE, CNNs, RNNs, RBM, NADE, AM, AN and DRL based recommender system. The deep learning technique in use determines the applicability of recommendation model. For instance, MLP can easily model the non-linear interactions between users and items; CNNs are capable of extracting local and global representations from heterogeneous

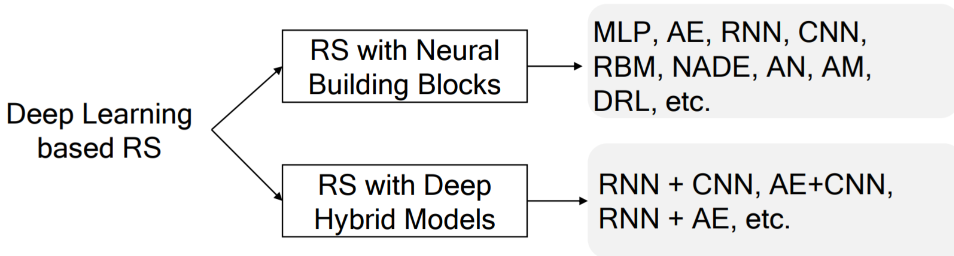


Fig. 1. Categories of deep neural network based recommendation models.

Table 1. A lookup table for reviewed publications.

Categories	Publications
MLP	[2, 13, 20, 27, 38, 47, 53, 54, 66, 92, 95, 157, 166, 185], [12, 39, 93, 112, 134, 154, 182, 183]
Autoencoder	[34, 88, 89, 114, 116, 125, 136, 137, 140, 159, 177, 187, 207], [4, 10, 32, 94, 150, 151, 158, 170, 171, 188, 196, 208, 209]
CNNs	[25, 49, 50, 75, 76, 98, 105, 127, 130, 153, 165, 172, 202, 206], [6, 44, 51, 83, 110, 126, 143, 148, 169, 190, 191]
RNNs	[5, 28, 35, 56, 57, 73, 78, 90, 117, 132, 139, 142, 174–176], [24, 29, 33, 55, 68, 91, 108, 113, 133, 141, 149, 173, 179]
RBM	[42, 71, 72, 100, 123, 167, 180]
NADE	[36, 203, 204]
Neural Attention	[14, 44, 70, 90, 99, 101, 127, 145, 169, 189, 194, 205], [62, 146, 193]
Adversary Network	[9, 52, 162, 164]
DRL	[16, 21, 107, 168, 198–200]
Hybrid Models	[17, 38, 41, 82, 84, 87, 118, 135, 160, 192, 193]

sources such as textual and visual information; RNNs enable the recommender system to model the temporal dynamics and sequential evolution of content information.

- *Recommendation with Deep Hybrid Models.* Some deep learning based recommendation models utilize more than one deep learning technique. The flexibility of deep neural networks makes it possible to combine several neural building blocks together to complement one another and form a more powerful hybrid model. There are many possible combinations of these night deep learning techniques but not all have been exploited. Note that it is different from the hybrid deep networks in [31] which refer to the deep architectures that make use of both generative and discriminative components.

Table 1 lists all the reviewed models, we organize them following the aforementioned classification scheme. Additionally, we also summarize some of the publications from the task perspective in Table 2. The reviewed publications are concerned with a variety of tasks. Some of the tasks have started to gain attention due to use of deep neural networks such as session-based recommendation, image, video recommendations. Some of the tasks might not be novel to the recommendation research area (a detail review on the side information for recommender systems can be found in [131]), but DL provides more possibility to find better solutions. For example, dealing with images and videos would be tough task without the help of deep learning techniques. The

Table 2. Deep neural network based recommendation models in specific application fields.

Data Sources/Tasks	Notes	Publications
Sequential Information	w/t User ID	[16, 29, 33, 35, 73, 91, 117, 133, 143, 160, 173, 175, 189, 194, 198, 205]
	Session based w/o User ID	[55–57, 68, 73, 99, 101, 102, 117, 142, 148, 149]
	Check-In, POI	[150, 151, 165, 185]
Text	Hash Tags	[44, 110, 118, 158, 182, 183, 193, 209]
	News	[10, 12, 113, 135, 169, 200]
	Review texts	[11, 87, 126, 146, 174, 197, 202]
	Quotes	[82, 141]
Images	Visual features	[2, 14, 25, 49, 50, 84, 98, 105, 112, 165, 172, 179, 191, 192, 197, 206]
Audio	Music	[95, 153, 167, 168]
Video	Videos	[14, 17, 27, 83]
Networks	Citation Network	[9, 38, 66]
	Social Network	[32, 116, 166]
	Cross Domain	[39, 92, 166]
Others	Cold-start	[154, 156, 170, 171]
	Multitask	[5, 73, 87, 174, 187]
	Explainability	[87, 126]

sequence modelling capability of deep neural networks makes it easy to capture the sequential patterns of user behaviors. Some of the specific tasks will be discussed in the following text.

3.2 Multilayer Perceptron based Recommendation

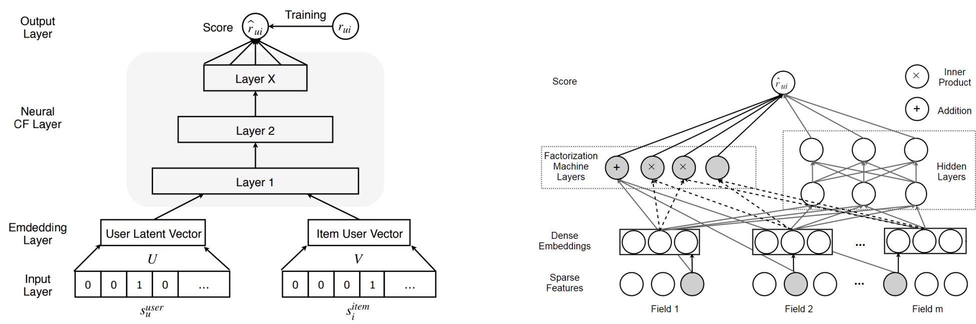
MLP is a concise but effective network which has been demonstrated to be able to approximate any measurable function to any desired degree of accuracy [59]. As such, it is the basis of numerous advanced approaches and is widely used in many areas.

Neural Extension of Traditional Recommendation Methods. Many existing recommendation models are essentially linear methods. MLP can be used to add nonlinear transformation to existing RS approaches and interpret them into neural extensions.

Neural Collaborative Filtering. In most cases, recommendation is deemed to be a two-way interaction between users preferences and items features. For example, matrix factorization decomposes the rating matrix into low-dimensional user/item latent factors. It is natural to construct a dual neural network to model the two-way interaction between users and items. Neural Network Matrix Factorization (NNMF) [37] and Neural Collaborative Filtering (NCF) [53] are two representative works. Figure 2a shows the NCF architecture. Let s_u^{user} and s_i^{item} denote the side information (e.g. user profiles and item features), or just one-hot identifier of user u and item i. The scoring function is defined as follows:

$$\hat{r}_{ui} = f(U^T \cdot s_u^{user}, V^T \cdot s_i^{item} | U, V, \theta) \tag{1}$$

where function $f(\cdot)$ represents the multilayer perceptron, and θ is the parameters of this network. Traditional MF can be viewed as a special case of NCF. Therefore, it is convenient to fuse the neural interpretation of matrix factorization with MLP to formulate a more general model which makes use of both linearity of MF and non-linearity of MLP to enhance recommendation quality. The whole network can be trained with weighted



(a) (b)

Fig. 2. Illustration of: (a) Neural Collaborative Filtering; (b) Deep Factorization Machine.

square loss (for explicit feedback) or binary cross-entropy loss (for implicit feedback). The cross-entropy loss is defined as:

$$\mathcal{L} = -\sum_{(u,i)\in\mathcal{O}\cup\mathcal{O}^-} r_{ui}\log\hat{r}_{ui} + (1-r_{ui})\log(1-\hat{r}_{ui}) \tag{2}$$

Negative sampling approaches can be used to reduce the number of training unobserved instances. Follow-up work [112, 134] proposed using pairwise ranking loss to enhance the performance. He et al. [92, 166] extended the NCF model to cross-domain recommendations. Xue et al. [184] and Zhang et al. [195] showed that the one-hot identifier can be replaced with columns or rows of the interaction matrix to retain the user-item interaction patterns.

Deep Factorization Machine. DeepFM [47] is an end-to-end model which seamlessly integrates factorization machine and MLP. It is able to model the high-order feature interactions via deep neural network and low-order interactions with factorization machine. Factorization machine (FM) utilizes addition and inner product operations to capture the linear and pairwise interactions between features (refer to Equation (1) in [119] for more details). MLP leverages the non-linear activations and deep structure to model the high-order interactions. The way of combining MLP with FM is enlightened by wide & deep network. It replaces the wide component with a neural interpretation of factorization machine. Compared to wide & deep model, DeepFM does not require tedious feature engineering. Figure 2b illustrates the structure of DeepFM. The input of DeepFM x is an m-fields data consisting of pairs (u, i) (identity and features of user and item). For simplicity, the outputs of FM and MLP are denoted as $y_{FM}(x)$ and $y_{MLP}(x)$ respectively. The prediction score is calculated by:

$$\hat{r}_{ui} = \sigma(y_{FM}(x) + y_{MLP}(x)) \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid activation function.

Lian et al. [93] improved DeepMF by proposing a eXtreme deep factorization machine to jointly model the explicit and implicit feature interactions. The explicit high-order feature interactions are learned via a compressed interaction network. A parallel work proposed by He et al. [54] replaces the second-order interactions with MLP and proposed regularizing the model with dropout and batch normalization.

Feature Representation Learning with MLP. Using MLP for feature representation is very straightforward and highly efficient, even though it might not be as expressive as autoencoder, CNNs and RNNs.

Wide & Deep Learning. This general model (shown in Figure 3a) can solve both regression and classification problems, but initially introduced for App recommendation in Google play [20]. The wide learning component is a single layer perceptron which can also be regarded as a generalized linear model. The deep learning component is multilayer perceptron. The rationale of combining these two learning techniques is that it enables the recommender to capture both memorization and generalization. Memorization achieved by the wide learning component represents the capability of catching the direct features from historical data. Meanwhile, the deep learning component catches the generalization by producing more general and abstract representations. This model can improve the accuracy as well as the diversity of recommendation.

Formally, the wide learning is defined as: $y = W_{wide}^T \{x, \phi(x)\} + b$, where W_{wide}^T, b are the model parameters. The input $\{x, \phi(x)\}$ is the concatenated feature set consisting of raw input feature x and transformed (e.g. cross-product transformation to capture the correlations between features) feature $\phi(x)$. Each layer of the deep neural component is in the form of $\alpha^{(l+1)} = f(W_{deep}^{(l)} a^{(l)} + b^{(l)})$, where l indicates the l^{th} layer, and $f(\cdot)$ is the activation function. $W_{deep}^{(l)}$ and $b^{(l)}$ are weight and bias terms. The wide & deep learning model is attained by fusing these two models:

$$P(\hat{r}_{ui} = 1|x) = \sigma(W_{wide}^T \{x, \phi(x)\} + W_{deep}^T a^{(l_f)} + bias) \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function, $\hat{r}_{ui}$ is the binary rating label, $a^{(l_f)}$ is the final activation. This joint model is optimized with stochastic back-propagation (follow-the-regularized-leader algorithm). Recommending list is generated based on the predicted scores.

By extending this model, Chen et al. [13] devised a locally-connected wide & deep learning model for large scale industrial-level recommendation task. It employs the efficient locally-connected network to replace the deep learning component, which decreases the running time by one order of magnitude. An important step of deploying wide & deep learning is selecting features for wide and deep parts. In other word, the system should be able to determine which features are memorized or generalized. Moreover, the cross-product transformation also is required to be manually designed. These pre-steps will greatly influence the utility of this model. The above mentioned deep factorization based model can alleviate the effort in feature engineering.

Covington et al. [27] explored applying MLP in YouTube recommendation. This system divides the recommendation task into two stages: candidate generation and candidate ranking. The candidate generation network retrieves a subset (hundreds) from all video corpus. The ranking network generates a top-n list (dozens) based on the nearest neighbors scores from the candidates. We notice that the industrial world cares more about feature engineering (e.g. transformation, normalization, crossing) and scalability of recommendation models.

Alashkar et al. [2] proposed a MLP based model for makeup recommendation. This work uses two identical MLPs to model labeled examples and expert rules respectively. Parameters of these two networks are updated simultaneously by minimizing the differences between their outputs. It demonstrates the efficacy of adopting expert knowledge to guide the learning process of the recommendation model in a MLP framework. It is highly precise even though the expertise acquisition needs a lot of human involvements.

Collaborative Metric Learning (CML). CML [60] replaces the dot product of MF with Euclidean distance because dot product does not satisfy the triangle inequality of distance function. The user and item embeddings are learned via maximizing the distance between users and their disliked items and minimizing that between users and their preferred items. In CML, MLP is used to learn representations from item features such as text, images and tags.

Recommendation with Deep Structured Semantic Model. Deep Structured Semantic Model (DSSM) [65] is a deep neural network for learning semantic representations of entities in a common continuous semantic space and measuring their semantic similarities. It is widely used in information retrieval area and is supremely suitable for top-n recommendation [39, 182]. DSSM projects different entities into a common low-dimensional space, and computes their similarities with cosine function. Basic DSSM is made up of MLP so we put it in this section. Note that, more advanced neural layers such as convolution and max-pooling layers can also be easily integrated into DSSM.

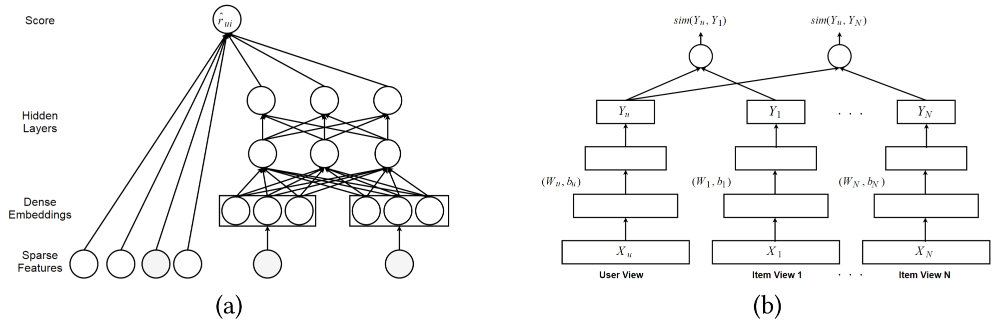


Fig. 3. Illustration of: (a) Wide & Deep Learning; (b) Multi-View Deep Neural Network.

Deep Semantic Similarity based Personalized Recommendation (DSPR) [182] is a tag-aware personalized recommender where each user x_u and item x_i are represented by tag annotations and mapped into a common tag space. Cosine similarity $sim(u, i)$ are applied to decide the relevance of items and users (or user's preference over the item). The loss function of DSPR is defined as follows:

$$\mathcal{L} = -\sum_{(u,i*)}[log(e^{sim(u,i*)}) - log(\sum_{(u,i^-)\in D^-} e^{sim(u,i^-)})] \quad (5)$$

where (u, i^-) are negative samples which are randomly sampled from the negative user item pairs. The authors. [183] further improved DSPR using autoencoder to learn low-dimensional representations from user/item profiles.

Multi-View Deep Neural Network (MV-DNN) [39] is designed for cross domain recommendation. It treats users as the pivot view and each domain (suppose we have Z domains) as auxiliary view. Apparently, there are Z similarity scores for Z user-domain pairs. Figure 3b illustrates the structure of MV-DNN. The loss function of MV-DNN is defined as:

$$\mathcal{L} = \underset{\theta}{argmin} \sum_{j=1}^{Z} \frac{exp(\gamma \cdot cosine(Y_u, Y_{a,j}))}{\sum_{X' \in R^{da}} exp(\gamma \cdot cosine(Y_u, f_a(X')))} \quad (6)$$

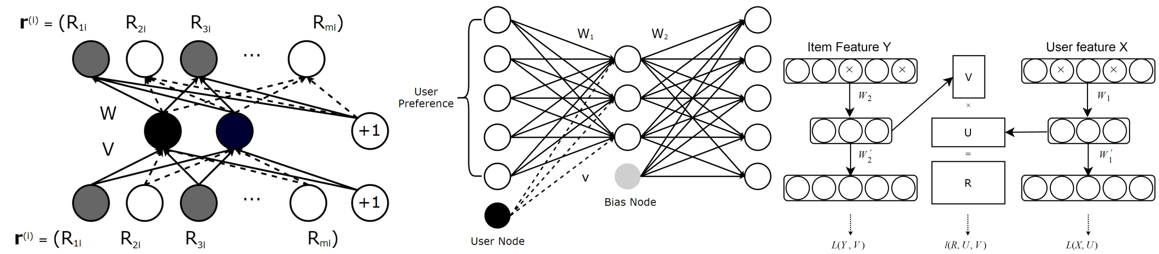
where θ is the model parameters, γ is the smoothing factor, Y_u is the output of user view, a is the index of active view. R^{da} is the input domain of view a. MV-DNN is capable of scaling up to many domains. However, it is based on the hypothesis that users have similar tastes in one domain should have similar tastes in other domains. Intuitively, this assumption might be unreasonable in many cases. Therefore, we should have some preliminary knowledge on the correlations across different domains to make the most of MV-DNN.

3.3 Autoencoder based Recommendation

There exist two general ways of applying autoencoder to recommender system: (1) using autoencoder to learn lower-dimensional feature representations at the bottleneck layer; or (2) filling the blanks of the interaction matrix directly in the reconstruction layer. Almost all the autoencoder variants such as denoising autoencoder, variational autoencoder, contactive autoencoder and marginalized autoencoder can be applied to recommendation task. Table 3 summarizes the recommendation models based on the types of autoencoder in use.

Autoencoder based Collaborative Filtering. One of the successful application is to consider the collaborative filtering from Autoencoder perspective.

AutoRec [125] takes user partial vectors $\mathbf{r}^{(u)}$ or item partial vectors $\mathbf{r}^{(i)}$ as input, and aims to reconstruct them in the output layer. Apparently, it has two variants: item-based AutoRec (I-AutoRec) and user-based AutoRec



(a) (b) (c)

Fig. 4. Illustration of: (a) Item based AutoRec; (b) Collaborative denoising autoencoder; (c) Deep collaborative filtering framework.

Table 3. Summary of four autoencoder based recommendation models

Vanilla/Denoising AE	Variational AE	Contractive AE	Marginalized AE
[114, 125, 136, 137, 159, 177] [70, 116, 170, 171, 188]	[19, 89, 94]	[196]	[88]

(U-AutoRec), corresponding to the two types of inputs. Here, we only introduce I-AutoRec, while U-AutoRec can be easily derived accordingly. Figure 4a illustrates the structure of I-AutoRec. Given input $\mathbf{r}^{(i)}$, the reconstruction is: $h(\mathbf{r}^{(i)};\theta) = f(W \cdot g(V \cdot \mathbf{r}^{(i)} + \mu) + b)$, where $f(\cdot)$ and $g(\cdot)$ are the activation functions, parameter $\theta = \{W, V, \mu, b\}$. The objective function of I-AutoRec is formulated as follows:

$$\underset{\theta}{argmin} \sum_{i=1}^{N} \parallel \mathbf{r}^{(i)} - h(\mathbf{r}^{(i)};\theta) \parallel_{\mathcal{O}}^2 + \lambda \cdot reg \tag{7}$$

Here $\| \cdot \|_{\mathcal{O}}^2$ means that it only considers observed ratings. The objective function can be optimized by resilient propagation (converges faster and produces comparable results) or L-BFGS (Limited-memory Broyden Fletcher Goldfarb Shanno algorithm). There are four important points about AutoRec that worth noticing before deployment: (1) I-AutoRec performs better than U-AutoRec, which may be due to the higher variance of user partially observed vectors. (2) Different combination of activation functions $f(\cdot)$ and $g(\cdot)$ will influence the performance considerably. (3) Increasing the hidden unit size moderately will improve the result as expanding the hidden layer dimensionality gives AutoRec more capacity to model the characteristics of the input. (4) Adding more layers to formulate a deep network can lead to slightly improvement.

CFN [136, 137] is an extension of AutoRec, and posses the following two advantages: (1) it deploys the denoising techniques, which makes CFN more robust; (2) it incorporates the side information such as user profiles and item descriptions to mitigate the sparsity and cold start influence. The input of CFN is also partial observed vectors, so it also has two variants: I-CFN and U-CFN, taking $\mathbf{r}^{(i)}$ and $\mathbf{r}^{(u)}$ as input respectively. Masking noise is imposed as a strong regularizer to better deal with missing elements (their values are zero). The authors introduced three widely used corruption approaches to corrupt the input: Gaussian noise, masking noise and salt-and-pepper noise. Further extension of CFN also incorporates side information. However, instead of just integrating side information in the first layer, CFN injects side information in every layer. Thus, the reconstruction becomes:

$$h(\{\tilde{\mathbf{r}}^{(i)}, \mathbf{s}_i\}) = f(W_2 \cdot \{g(W_1 \cdot \{\mathbf{r}^{(i)}, \mathbf{s}_i\} + \mu), \mathbf{s}_i\} + b) \tag{8}$$

where $\mathbf{s}_i$ is side information, $\{\tilde{\mathbf{r}}^{(i)}, \mathbf{s}_i\}$ indicates the concatenation of $\tilde{\mathbf{r}}^{(i)}$ and $\mathbf{s}_i$. Incorporating side information improves the prediction accuracy, speeds up the training process and enables the model to be more robust.

Collaborative Denoising Auto-Encoder (CDAE). The three models reviewed earlier are mainly designed for rating prediction, while CDAE [177] is principally used for ranking prediction. The input of CDAE is user partially observed implicit feedback $\mathbf{r}^{(u)}_{pref}$. The entry value is 1 if the user likes the movie, otherwise 0. It can also be regarded as a preference vector which reflects user's interests to items. Figure 4b illustrates the structure of CDAE. The input of CDAE is corrupted by Gaussian noise. The corrupted input $\tilde{\mathbf{r}}^{(u)}_{pref}$ is drawn from a conditional Gaussian distribution $p(\tilde{\mathbf{r}}^{(u)}_{pref}|\mathbf{r}^{(u)}_{pref})$. The reconstruction is defined as:

$$h(\tilde{\mathbf{r}}^{(u)}_{pref}) = f(W_2 \cdot g(W_1 \cdot \tilde{\mathbf{r}}^{(u)}_{pref} + V_u + b_1) + b_2) \tag{9}$$

where $V_u \in \mathbb{R}^K$ denotes the weight matrix for user node (see figure 4b). This weight matrix is unique for each user and has significant influence on the model performance. Parameters of CDAE are also learned by minimizing the reconstruction error:

$$\underset{W_1,W_2,V,b_1,b_2}{argmin} \frac{1}{M} \sum_{u=1}^{M} \mathbf{E}_{p(\tilde{\mathbf{r}}^{(u)}_{pref}|\mathbf{r}^{(u)}_{pref})}[\ell(\tilde{\mathbf{r}}^{(u)}_{pref}, h(\tilde{\mathbf{r}}^{(u)}_{pref}))] + \lambda \cdot reg \tag{10}$$

where the loss function $\ell(\cdot)$ can be square loss or logistic loss.

CDAE initially updates its parameters using SGD over all feedback. However, the authors argued that it is impractical to take all ratings into consideration in real world applications, so they proposed a negative sampling technique to sample a small subset from the negative set (items with which the user has not interacted), which reduces the time complexity substantially without degrading the ranking quality.

Multi-VAE and Multi-DAE [94] proposed a variant of varitional autoencoder for recommendation with implicit data, showing better performance than CDAE. The authors introduced a principled Bayesian inference approach for parameters estimation and show favorable results than commonly used likelihood functions.

To the extent of our knowledge, Autoencoder-based Collaborative Filtering (ACF) [114] is the first autoencoder based collaborative recommendation model. Instead of using the original partial observed vectors, it decomposes them by integer ratings. For example, if the rating score is integer in the range of [1-5], each $\mathbf{r}^{(i)}$ will be divided into five partial vectors. Similar to AutoRec and CFN, the cost function of ACF aims at reducing the mean squared error. However, there are two demerits of ACF: (1) it fails to deal with non-integer ratings; (2) the decomposition of partial observed vectors increases the sparseness of input data and leads to worse prediction accuracy.

Feature Representation Learning with Autoencoder. Autoencoder is a class of powerful feature representation learning approach. As such, it can also be used in recommender systems to learn feature representations from user/item content features.

Collaborative Deep Learning (CDL). CDL [159] is a hierarchical Bayesian model which integrates stacked denoising autoencoder (SDAE) into probabilistic matrix factorization. To seamlessly combine deep learning and recommendation model, the authors proposed a general Bayesian deep learning framework [161] consisting of two tightly hinged components: perception component (deep neural network) and task-specific component. Specifically, the perception component of CDL is a probabilistic interpretation of ordinal SDAE, and PMF acts as the task-specific component. This tight combination enables CDL to balance the influences of side information and interaction history. The generative process of CDL is as follows:

(1) For each layer l of the SDAE: (a) For each column n of weight matrix W_l, draw $W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{D_l})$; (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{D_l})$; (c) For each row i of X_l, draw $X_{l,i*} \sim \mathcal{N}(\sigma(X_{l-1,i*}W_l + b_l), \lambda_s^{-1}\mathbf{I}_{D_l})$.
(2) For each item i: (a) Draw a clean input $X_{c,i*} \sim \mathcal{N}(X_{L,i*}, \lambda_n^{-1}\mathbf{I}_{I_i})$; (b) Draw a latent offset vector $\epsilon_i \sim \mathcal{N}(0, \lambda_v^{-1}\mathbf{I}_D)$ and set the latent item vector: $V_i = \epsilon_i + X^T_{\frac{L}{2},i*}$.
(3) Draw a latent user vector for each user u, $U_u \sim \mathcal{N}(0, \lambda_u^{-1}\mathbf{I}_D)$.
(4) Draw a rating r_{ui} for each user-item pair (u, i), $r_{ui} \sim \mathcal{N}(U_u^T V_i, C_{ui}^{-1})$.

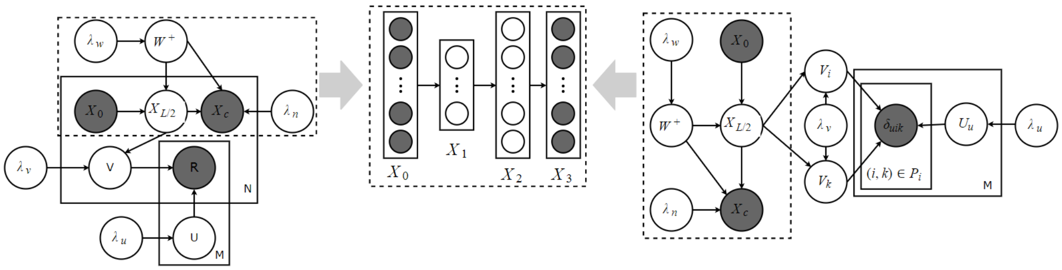


Fig. 5. Graphical model of collaborative deep learning (left) and collaborative deep ranking (right).

where W_l and b_l are the weight matrix and biases vector for layer l, X_l represents layer l. $\lambda_w, \lambda_s, \lambda_n, \lambda_v, \lambda_u$ are hyper-parameters, C_{ij} is a confidence parameter for determining the confidence to observations [63]. Figure 5(left) illustrates the graphical model of CDL. The authors exploited an EM-style algorithm to learn the parameters. In each iteration, it updates U and V first, and then updates W and b by fixing U and V. The authors also introduced a sampling-based algorithm [161] to avoid the local optimum.

Before CDL, Wang et al. [158] proposed a similar model, relational stacked denoising autoencoders (RSDAE), for tag recommendation. The difference of CDL and RSDAE is that RSDAE replaces the PMF with a relational information matrix. Another extension of CDL is collaborative variational autoencoder (CVAE) [89], which replaces the deep neural component of CDL with a variational autoencoder. CVAE learns probabilistic latent variables for content information and can easily incorporate multimedia (video, images) data sources.

Collaborative Deep Ranking (CDR). CDR [188] is devised specifically in a pairwise framework for top-n recommendation. Some studies have demonstrated that pairwise model is more suitable for ranking lists generation [120, 177, 188]. Experimental results also show that CDR outperforms CDL in terms of ranking prediction. Figure 5(right) presents the structure of CDR. The first and second generative process steps of CDR are the same as CDL. The third and fourth steps are replaced by the following step:

- For each user u: (a) Draw a latent user vector for u, $U_u \sim \mathcal{N}(0, \lambda_u^{-1}\mathbf{I}_D)$; (b) For each pair-wise preference $(i,j) \in P_i$, where $P_i = \{(i,j) : r_{ui} - r_{uj} > 0\}$, draw the estimator, $\delta_{uij} \sim \mathcal{N}(U_u^T V_i - U_u^T V_j, C_{uij}^{-1})$.

where $\delta_{uij} = r_{ui} - r_{uj}$ represents the pairwise relationship of user's preference on item i and item j, C_{uij}^{-1} is a confidence value which indicates how much user u prefers item i than item j. The optimization process is performed in the same manner as CDL.

Deep Collaborative Filtering Framework. It is a general framework for unifying deep learning approaches with collaborative filtering model [88]. This framework makes it easily to utilize deep feature learning techniques to build hybrid collaborative models. The aforementioned work such as [153, 159, 167] can be viewed as special cases of this general framework. Formally, the deep collaborative filtering framework is defined as follows:

$$\arg\min_{U,V} \ell(R,U,V) + \beta(\| U \|_F^2 + \| V \|_F^2) + \gamma\mathcal{L}(X,U) + \delta\mathcal{L}(Y,V) \tag{11}$$

where β, γ and δ are trade-off parameters to balance the influences of these three components, X and Y are side information, $\ell(\cdot)$ is the loss of collaborative filtering model. $\mathcal{L}(X,U)$ and $\mathcal{L}(Y,V)$ act as hinges for connecting deep learning and collaborative models and link side information with latent factors. On top of this framework, the authors proposed the marginalized denoising autoencoder based collaborative filtering model (mDA-CF). Compared to CDL, mDA-CF explores a more computationally efficient variants of autoencoder: marginalized denoising autoencoder [15]. It saves the computational costs for searching sufficient corrupted version of input by marginalizing out the corrupted input, which makes mDA-CF more scalable than CDL. In addition, mDA-CF embeds content information of items and users while CDL only considers the effects of item features.

AutoSVD++ [196] makes use of contractive autoencoder [122] to learn item feature representations, then integrates them into the classic recommendation model, SVD++ [79]. The proposed model posses the following advantages: (1) compared to other autoencoders variants, contractive autoencoder captures the infinitesimal input variations; (2) it models the implicit feedback to further enhance the accuracy; (3) an efficient training algorithm is designed to reduce the training time.

HRCD [170, 171] is a hybrid collaborative model based on autoencoder and timeSVD++ [80]. It is a time-aware model which uses SDAE to learn item representations from raw features and aims at solving the cold item problem.

3.4 Convolutional Neural Networks based Recommendation

Convolution Neural Networks are powerful in processing unstructured multimedia data with convolution and pool operations. Most of the CNNs based recommendation models utilize CNNs for feature extraction.

Feature Representation Learning with CNNs. CNNs can be used for feature representation learning from multiple sources such as image, text, audio, video, etc.

CNNs for Image Feature Extraction. Wang et al. [165] investigated the influences of visual features to Point-of-Interest (POI) recommendation, and proposed a visual content enhanced POI recommender system (VPOI). VPOI adopts CNNs to extract image features. The recommendation model is built on PMF by exploring the interactions between: (1) visual content and latent user factor; (2) visual content and latent location factor. Chu et al. [25] exploited the effectiveness of visual information (e.g. images of food and furnishings of the restaurant) in restaurant recommendation. The visual features extracted by CNN joint with the text representation are input into MF, BPRMF and FM to test their performance. Results show that visual information improves the performance to some degree but not significant. He et al. [50] designed a visual Bayesian personalized ranking (VBPR) algorithm by incorporating visual features (learned via CNNs) into matrix factorization. He et al. [49] extended VBPR with exploring user's fashion awareness and the evolution of visual factors that user considers when selecting items. Yu et al. [191] proposed a coupled matrix and tensor factorization model for aesthetic-based clothing recommendation, in which CNNs is used to learn the images features and aesthetic features. Nguyen et al. [110] proposed a personalized tag recommendation model based on CNNs. It utilizes the convolutional and max-pooling layer to get visual features from patches of images. User information is injected for generating personalized recommendation. To optimize this network, the BPR objective is adopted to maximize the differences between the relevant and irrelevant tags. Lei et al. [84] proposed a comparative deep leaning model with CNNs for image recommendation. This network consists of two CNNs which are used for image representation learning and a MLP for user preferences modelling. It compares two images (one positive image user likes and one negative image user dislikes) against a user. The training data is made up of triplets: t (user U_t, positive image I_t^+, negative image I_t^-). Assuming that the distance between user and positive image $D(\pi(U_t), \phi(I_t^+))$ should be closer than the distance between user and negative images $D(\pi(U_t), \phi(I_t^-))$, where $D(\cdot)$ is the distance metric (e.g. Euclidean distance). ConTagNet [118] is a context-aware tag recommender system. The image features are learned by CNNs. The context representations are processed by a two layers fully-connected feedforward neural network. The outputs of two neural networks are concatenated and fed into a softmax funcation to predict the probability of candidate tags.

CNNs for Text Feature Extraction. DeepCoNN [202] adopts two parallel CNNs to model user behaviors and item properties from review texts. This model alleviates the sparsity problem and enhances the model interpretability by exploiting rich semantic representations of review texts with CNNs. It utilizes a word embedding technique to map the review texts into a lower-dimensional semantic space as well as keep the words sequences information.

The extracted review representations then pass through a convolutional layer with different kernels, a max-pooling layer, and a full-connected layer consecutively. The output of the user network x_u and item network x_i are finally concatenated as the input of the prediction layer where the factorization machine is applied to capture their interactions for rating prediction. Catherine et al. [11] mentioned that DeepCoNN only works well when the review text written by the target user for the target item is available at test time, which is unreasonable. As such, they extended it by introducing a latent layer to represent the target user-target-item pair. This model does not access the reviews during validation/test and can still remain good accuracy. Shen et al. [130] built an e-learning resources recommendation model. It uses CNNs to extract item features from text information of learning resources such as introduction and content of learning material, and follows the same procedure of [153] to perform recommendation. ConvMF [75] combines CNNs with PMF in a similar way as CDL. CDL uses autoencoder to learn the item feature representations, while ConvMF employs CNNs to learn high level item representations. The main advantage of ConvMF over CDL is that CNNs is able to capture more accurate contextual information of items via word embedding and convolutional kernels. Tuan et al. [148] proposed using CNNs to learn feature representations form item content information (e.g., name, descriptions, identifier and category) to enhance the accuracy of session based recommendation.

CNNs for Audio and Video Feature Extraction. Van et al. [153] proposed using CNNs to extract features from music signals. The convolutional kernels and pooling layers allow operations at multiple timescales. This content-based model can alleviate the cold start problem (music has not been consumed) of music recommendation. Lee et al. [83] proposed extracting audio features with the prominent CNNs model ResNet. The recommendation is performed in the collaborative metric learning framework similar to CML.

CNNs based Collaborative filtering. Directly applying CNNs to vanilla collaborative filtering is also viable. For example, He et al. [51] proposed using CNNs to improve NCF and presented the ConvNCF. It uses outer product instead of dot product to model the user item interaction patterns. CNNs are applied over the result of outer product and could capture the high-order correlations among embeddings dimensions. Tang et al. [143] presented sequential recommendation (with user identifier) with CNNs, where two CNNs (hierarchical and vertical) are used to model the union-level sequential patterns and skip behaviors for sequence-aware recommendation.

Graph CNNs for Recommendation. Graph convolutional Networks is a powerful tool for non-Eulcidean data such as: social networks, knowledge graphs, protein-interaction networks, etc [77]. Interactions in recommendation area can also be viewed as a such structured dataset (bipartite graph). Thus, it can also be applied to recommendation tasks. For example, Berg et al. [6] proposed considering the recommendation problem as a link prediction task with graph CNNs. This framework makes it easy to integrate user/item side information such as social networks and item relationships into recommendation model. Ying et al. [190] proposed using graph CNNs for recommendations in Pinterest[10]. This model generates item embeddings from both graph structure as well item feature information with random walk and graph CNNs, and is suitable for very large-scale web recommender. The proposed model has been deployed in Pinterest to address a variety of real-world recommendation tasks.

3.5 Recurrent Neural Networks based Recommendation

RNNs are extremely suitable for sequential data processing. As such, it becomes a natural choice for dealing with the temporal dynamics of interactions and sequential patterns of user behaviours, as well as side information with sequential signals, such as texts, audio, etc.

Session-based Recommendation without User Identifier. In many real world applications or websites, the system usually does not bother users to log in so that it has no access to user's identifier and her long period consumption habits or long-term interests. However, the session or cookie mechanisms enables those systems to

[10]https://www.pinterest.com

users's short term preferences. This is a relatively unappreciated task in recommender systems due to the extreme sparsity of training data. Recent advancements have demonstrated the efficacy of RNNs in solving this issue [56, 142, 176].

GRU4Rec. Hidasi et al. [56] proposed a session-based recommendation model, GRU4Rec, based GRU (shown in Figure 6a). The input is the actual state of session with 1-of-N encoding, where N is the number of items. The coordinate will be 1 if the corresponding item is active in this session, otherwise 0. The output is the likelihood of being the next in the session for each item. To efficiently train the proposed framework, the authors proposed a session-parallel mini-batches algorithm and a sampling method for output. The ranking loss which is also coined TOP1 and has the following form:

$$\mathcal{L}_s = \frac{1}{S}\sum_{j=1}^{S}\sigma(\hat{r}_{sj} - \hat{r}_{si}) + \sigma(\hat{r}_{sj}^2) \tag{12}$$

where S is the sample size, $\hat{r}_{si}$ and $\hat{r}_{sj}$ are the scores on negative item i and positive item j at session s, σ is the logistic sigmoid function. The last term is used as a regularization. Note that, BPR loss is also viable. A recent work [55] found that the original TOP1 loss and BPR loss defined in [56] suffer from the gradient vanishing problem, as such, two novel loss functions: TOP1-max and BPR-max are proposed.

The follow-up work [142] proposed several strategies to further improve this model: (1) augment the click sequences with sequence preprocessing and dropout regularization; (2) adapt to temporal changes by pre-training with full training data and fine-tuning the model with more recent click-sequences; (3) distillation the model with *privileged information* with a teacher model; (4) using item embedding to decrease the number of parameters for faster computation.

Wu et al. [176] designed a session-based recommendation model for real-world e-commerce website. It utilizes the basic RNNs to predict what user will buy next based on the click history. To minimize the computation costs, it only keeps a finite number of the latest states while collapsing the older states into a single history state. This method helps to balance the trade-off between computation costs and prediction accuracy. Quadrana et al. [117] presented a hierarchical recurrent neural network for session-based recommendation. This model can deal with both session-aware recommendation when user identifiers are present.

The aforementioned three session-based models do not consider any side information. Two extensions [57, 132] demonstrate that side information has effect on enhancing session recommendation quality. Hidasi et al. [57] introduced a parallel architecture for session-based recommendation which utilizes three GRUs to learn representations from identity one-hot vectors, image feature vectors and text feature vectors. The outputs of these three GRUs are weightedly concatenated and fed into a non-linear activation to predict the next items in that session. Smirnova et al. [132] proposed a context-aware session-based recommender system based on conditional RNNs. It injects context information into input and output layers. Experimental results of these two models suggest that models incorporated additional information outperform those solely based on historical interactions.

Despite the success of RNNs in session-based recommendation, Jannach et al. [68] indicated that simple neighbourhood approach could achieve same accuracy results as GRU4Rec. Combining the neighbourhood with RNNs methods can usually lead to best performance. This work suggests that some baselines in recent works are not well-justified and correctly evaluated. A more comprehensive discussion can be found in [103].

Sequential Recommendation with User Identifier. Unlike session-based recommender where user identifiers are usually not present. The following studies deal with the sequential recommendation task with known user identifications.

Recurrent Recommender Network (RRN) [175] is a non-parametric recommendation model built on RNNs (shown in Figure 6b). It is capable of modelling the seasonal evolution of items and changes of user preferences over time. RRN uses two LSTM networks as the building block to model dynamic user state u_{ut} and item state v_{it}. In the

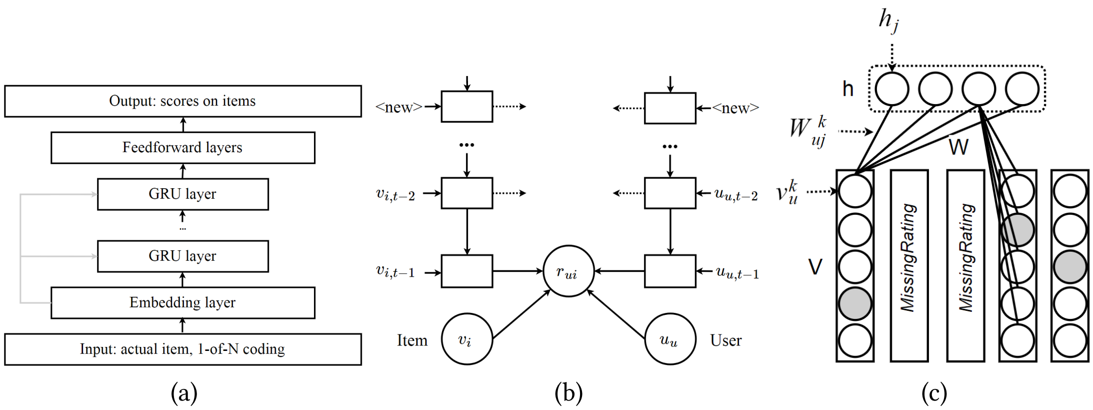


Fig. 6. Illustration of: (a) Session-based recommendation with RNN; (b) Recurrent recommender network; (c) Restricted Boltzmann Machine based Collaborative Filtering.

meantime, considering the fixed properties such as user long-term interests and item static features, the model also incorporates the stationary latent attributes of user and item: u_u and v_i. The predicted rating of item j given by user i at time t is defined as:

$$\hat{r}_{ui|t} = f(u_{ut}, v_{it}, u_u, v_i) \tag{13}$$

where u_{ut} and v_{it} are learned from LSTM, u_u and v_i are learned by the standard matrix factorization. The optimization is to minimize the square error between predicted and actual rating values.

Wu et al. [174] further improved the RRNs model by modelling text reviews and ratings simultaneously. Unlike most text review enhanced recommendation models [127, 202], this model aims to generate reviews with a character-level LSTM network with user and item latent states. The review generation task can be viewed as an auxiliary task to facilitate rating prediction. This model is able to improve the rating prediction accuracy, but cannot generate coherent and readable review texts. NRT [87] which will be introduced in the following text can generate readable review tips. Jing et al. [73] proposed a multi-task learning framework to simultaneously predict the returning time of users and recommend items. The returning time prediction is motivated by a survival analysis model designed for estimating the probability of survival of patients. The authors modified this model by using LSTM to estimate the returning time of costumers. The item recommendation is also performed via LSTM from user's past session actions. Unlike aforementioned session-based recommendations which focus on recommending in the same session, this model aims to provide inter-session recommendations. Li et al. [91] presented a behavior-intensive model for sequential recommendation. This model consists of two components: neural item embedding and discriminative behaviors learning. The latter part is made up of two LSTMs for session and preference behaviors learning respectively. Christakopoulou et al. [24] designed an interactive recommender with RNNs. The proposed framework aims to address two critical tasks in interactive recommender: ask and respond. RNNs are used to tackle both tasks: predict questions that the user might ask based on her recent behaviors(e.g, watch event) and predict the responses. Donkers et al. [35] designed a novel type of Gated Recurrent Unit to explicit represent individual user for next item recommendation.

Feature Representation Learning with RNNs. For side information with sequential patterns, using RNNs as the representation learning tool is an advisable choice.

Dai et al. [29] presented a co-evolutionary latent model to capture the co-evolution nature of users' and items' latent features. The interactions between users and items play an important role in driving the changes of user preferences and item status. To model the historical interactions, the author proposed using RNNs to automatically learn representations of the influences from drift, evolution and co-evolution of user and item features.

Bansal et al. [5] proposed using GRUs to encode the text sequences into latent factor model. This hybrid model solves both warm-start and cold-start problems. Furthermore, the authors adopted a multi-task regularizer to prevent overfitting and alleviate the sparsity of training data. The main task is rating prediction while the auxiliary task is item meta-data (e.g. tags, genres) prediction.

Okura et al. [113] proposed using GRUs to learn more expressive aggregation for user browsing history (browsed items), and recommend news articles with latent factor model. The results show a significant improvement compared with the traditional word-based approach. The system has been fully deployed to online production services and serving over ten million unique users everyday.

Li et al. [87] presented a multitask learning framework, NRT, for predicting ratings as well as generating textual tips for users simultaneously. The generated tips provide concise suggestions and anticipate user's experience and feelings on certain products. The rating prediction task is modelled by non-linear layers over item and user latent factors $U \in \mathbb{R}^{k_u \times M}$, $V \in \mathbb{R}^{k_v \times M}$, where k_u and k_v (not necessarily equal) are latent factor dimensions for users and items. The predicted rating r_{ui} and two latent factor matrices are fed into a GRU for tips generation. Here, r_{ui} is used as context information to decide the sentiment of the generated tips. The multi-task learning framework enables the whole model to be trained efficiently in an end-to-end paradigm.

Song et al. [135] designed a temporal DSSM model which integrates RNNs into DSSM for recommendation. Based on traditional DSSM, TDSSM replace the left network with item static features, and the right network with two sub-networks to modelling user static features (with MLP) and user temporal features (with RNNs).

3.6 Restricted Boltzmann Machine based Recommendation

Salakhutdinov et al. [123] proposed a restricted Boltzmann machine based recommender (shown in Figure 6c). To the best of our knowledge, it is the first recommendation model that built on neural networks. The visible unit of RBM is limited to binary values, therefore, the rating score is represented in a one-hot vector to adapt to this restriction. For example, [0,0,0,1,0] represents that the user gives a rating score 4 to this item. Let $h_j, j = 1, ..., F$ denote the hidden units with fixed size F. Each user has a unique RBM with shared parameters. Suppose a user rated m movies, the number of visible units is m, Let X be a $K \times m$ matrix where $x_i^y = 1$ if user u rated movie i as y and $x_i^y = 0$ otherwise. Then:

$$p(v_i^y = 1|h) = \frac{exp(b_i^y + \sum_{j=1}^F h_j W_{ij}^y)}{\sum_{l=1}^K exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)} \quad , \quad p(h_j = 1|X) = \sigma(b_j + \sum_{i=1}^m \sum_{y=1}^K x_i^y W_{ij}^y) \tag{14}$$

where W_{ij}^y represents the weight on the connection between the rating y of movie i and the hidden unit j, b_i^y is the bias of rating y for movie i, b_j is the bias of hidden unit j. RBM is not tractable, but the parameters can be learned via the Contrastive Divergence (CD) algorithm [45]. The authors further proposed using a conditional RBM to incorporate the implicit feedback. The essence here is that users implicitly tell their preferences by giving ratings, regardless of how they rate items.

The above RBM-CF is user-based where a given user's rating is clamped on the visible layer. Similarity, we can easily design an item-based RBM-CF if we clamp a given item's rating on the visible layer. Georgiev et al. [42] proposed to combine the user-based and item-based RBM-CF in a unified framework. In the case, the visible units are determined both by user and item hidden units. Liu et al. [100] designed a hybrid RBM-CF which incorporates item features (item categories). This model is also based on conditional RBM. There are two differences between this hybrid model with the conditional RBM-CF with implicit feedback: (1) the conditional layer here is modelled

Table 4. Categories of neural attention based recommendation models.

Vanilla Attention	Co-Attention
[14, 44, 70, 90, 99, 101, 127, 145, 169, 189]	[62, 146, 193, 194, 205]

for the binary item genres; (2) the conditional layer affects both the hidden layer and the visible layer with different connected weights.

3.7 Neural Attention based Recommendation

Attention mechanism is motivated by human visual attention. For example, people only need to focus on specific parts of the visual inputs to understand or recognize them. Attention mechanism is capable of filtering out the uninformative features from raw inputs and reduce the side effects of noisy data. It is an intuitive but effective technique and has garnered considerable attention over the recent years across areas such as computer vision [3], natural language processing [104, 155] and speech recognition [22, 23]. Neural attention can not only used in conjunction with MLP, CNNs and RNNs, but also address some tasks independently [155]. Integrating attention mechanism into RNNs enables the RNNs to process long and noisy inputs [23]. Although LSTM can solve the long memory problem theoretically, it is still problematic when dealing with long-range dependencies. Attention mechanism provides a better solution and helps the network to better memorize inputs. Attention-based CNNs are capable of capturing the most informative elements of the inputs [127]. By applying attention mechanism to recommender system, one could leverage attention mechanism to filter out uninformative content and select the most representative items [14] while providing good interpretability. Although neural attention mechanism is not exactly a standalone deep neural technique, it is still worthwhile to discuss it separately due to its widespread use.

Attention model learns to attend to the input with attention scores. Calculating the attention scores lives at the heart of neural attention models. Based on the way for calculating the attention scores, we classify the neural attention models into (1) standard vanilla attention and (2) co-attention. Vanilla attention utilizes a parameterized context vector to learn to attend while co-attention is concerned with learning attention weights from two-sequences. Self-attention is a special case of co-attention. Recent works [14, 44, 127] demonstrate the capability of attention mechanism in enhancing recommendation performance. Table 4 summarizes the attention based recommendation models.

Recommendation with Vanilla Attention

Chen et al. [14] proposed an attentive collaborative filtering model by introducing a two-level attention mechanism to latent factor model. It consists of item-level and component-level attention. The item-level attention is used to select the most representative items to characterize users. The component-level attention aims to capture the most informative features from multimedia auxiliary information for each user. Tay et al. [145] proposed a memory-based attention for collaborative metric learning. It introduces a latent relation vector learned via attention to CML. Jhamb et al. [70] proposed using attention mechanism to improve the performance of autoencoder based CF. Liu et al. [99] proposed a short-term attention and memory priority based model, in which both long and short term user interests are intergrated for session based recommendation. Ying et al. [189] proposed a hierarchical attention model for sequential recommendation. Two attention networks are used to model user long-term and short-term interests.

Introducing attention mechanism to RNNs could significantly improve their performance. Li et al. [90] proposed such an attention-based LSTM model for hashtag recommendation. This work takes the advantages of both RNNs and attention mechanism to capture the sequential property and recognize the informative words from microblog posts. Loyala et al. [101] proposed an encoder-decoder architecture with attention for user session and intents modelling. This model consists of two RNNs and could capture the transition regularities in a more expressive way.

Vanilla attention can also work in conjunction with CNNs for recommender tasks. Gong et al. [44] proposed an attention based CNNs system for hashtag recommendation in microblog. It treats hashtag recommendation as a multi-label classification problem. The proposed model consists of a global channel and a local attention channel. The global channel is made up of convolution filters and max-pooling layers. All words are encoded in the input of global channel. The local attention channel has an attention layer with given window size and threshold to select informative words (known as trigger words in this work). Hence, only trigger words are at play in the subsequent layers. In the follow-up work [127], Seo et al. made use of two neural networks same as [44] (without the last two layers) to learn feature representations from user and item review texts, and predict rating scores with dot product in the final layer. Wang et al. [169] presented a combined model for article recommendation, in which CNNs is used to learn article representations and attention is utilized to deal with the diverse variance of editors's selection behavior.

Recommendation with Co-Attention Zhang et al. [194] proposed a combined model, AttRec, which improves the sequential recommendation performance by capitalizing the strength of both self-attention and metric learning. It uses self-attention to learn user short-term intents from her recent interactions and takes the advantages of metric learning to learn more expressive user and item embemddings. Zhou et al. [205] proposed using self-attention for user heterogeneous behaviour modelling. Self-attention is simple yet effective mechanism and has shown superior performance than CNNs and RNNs in terms of sequential recommendation task. We believe that it has the capability to replace many complex neural models and more investigation is expected. Tay et al. [146] proposed a review based recommendation system with multi-pointer co-attention. The co-attention enables the model to select information reviews via co-learning from both user and item reviews. Zhang et al. [193] proposed a co-atention based hashtag recommendation model that integrates both visual and textual information. Shi et al. [62] proposed a neural co-attention model for personalized ranking task with meta-path.

3.8 Neural AutoRegressive based Recommendation

As mentioned above, RBM is not tractable, thus we usually use the Contrastive Divergence algorithm to approximate the log-likelihood gradient on the parameters [81], which also limits the usage of RBM-CF. The so-called Neural Autoregressive Distribution Estimator (NADE) is a tractable distribution estimator which provides a desirable alternative to RBM. Inspired by RBM-CF, Zheng et al. [204] proposed a NADE based collaborative filtering model (CF-NADE). CF-NADE models the distribution of user ratings. Here, we present a detailed example to illustrate how the CF-NADE works. Suppose we have 4 movies: m1 (rating is 4), m2 (rating is 2), m3 (rating is 3) and m4 (rating is 5). The CF-NADE models the joint probability of the rating vector r by the chain rule: $p(\mathbf{r}) = \prod_{i=1}^{D} p(r_{m_{o_i}}|\mathbf{r}_{m_{o_{<i}}})$, where D is the number of items that the user has rated, o is the D-tuple in the permutations of $(1, 2, ..., D)$, m_i is the index of the i^{th} rated item, $r_{m_{o_i}}$ is the rating that the user gives to item m_{o_i}. More specifically, the procedure goes as follows: (1) the probability that the user gives $m1$ 4-star conditioned on nothing; (2) the probability that the user gives $m2$ 2-star conditioned on giving $m1$ 4-star; (3) the probability that the user gives $m3$ 3-star conditioned on giving $m1$ 4-star and $m2$ 2-star; (4) the probability that the user gives $m4$ 5-star conditioned on giving $m1$ 4-star, $m2$ 2-star and $m3$ 3-star.

Ideally, the order of movies should follow the time-stamps of ratings. However, empirical study shows that random drawing also yields good performances. This model can be further extended to a deep model. In the follow-up paper, Zheng et al. [203] proposed incorporating implicit feedback to overcome the sparsity problem of rating matrix. Du et al. [36] further imporved this model with a user-item co-autoregressive approach, which achieves better performance in both rating estimation and personalized ranking tasks.

3.9 Deep Reinforcement Learning for Recommendation

Most recommendation models consider the recommendation process as a static process, which makes it difficult to capture user's temporal intentions and to respond in a timely manner. In recent years, DRL has begun to garner attention [21, 107, 168, 198–200] in making personalized recommendation. Zhao et al. [199] proposed a DRL framework, DEERS, for recommendation with both negative and positive feedback in a sequential interaction setting. Zhao et al. [198] explored the page-wise recommendation scenario with DRL, the proposed framework DeepPage is able to adaptively optimize a page of items based on user's real-time actions. Zheng et al. [200] proposed a news recommendation system, DRN, with DRL to tackle the following three challenges: (1) dynamic changes of news content and user preference; (2) incorporating return patterns (to the service) of users; (3) increase diversity of recommendations. Chen et al. [16] proposed a robust deep Q-learning algorithm to address the unstable reward estimation issue with two strategies: stratified sampling replay and approximate regretted reward. Choi et al. [21] proposed solving the cold-start problem with RL and bi-clustering. Munemasa et al [107] proposed using DRL for stores recommendation.

Reinforcement Learning techniques such as contextual-bandit approach [86] had shown superior recommendation performance in real-world applications. Deep neural networks increase the practicality of RL and make it possible to model various of extra information for designing real-time recommendation strategies.

3.10 Adversarial Network based Recommendation

IRGAN [162] is the first model which applies GAN to information retrieval area. Specifically, the authors demonstrated its capability in three information retrieval tasks, including: web search, item recommendation and question answering. In this survey, we mainly focus on how to use IRGAN to recommend items.

Firstly, we introduce the general framework of IRGAN. Traditional GAN consists of a discriminator and a generator. Likely, there are two schools of thinking in information retrieval, that is, generative retrieval and discriminative retrieval. Generative retrieval assumes that there is an underlying generative process between documents and queries, and retrieval tasks can be achieved by generating relevant document d given a query q. Discriminative retrieval learns to predict the relevance score r given labelled relevant query-document pairs. The aim of IRGAN is to combine these two thoughts into a unified model, and make them to play a minimax game like generator and discriminator in GAN. The generative retrieval aims to generate relevant documents similar to ground truth to fool the discriminative retrieval model.

Formally, let $p_{true}(d|q_n, r)$ refer to the user's relevance (preference) distribution. The generative retrieval model $p_\theta(d|q_n, r)$ tries to approximate the true relevance distribution. Discriminative retrieval $f_\phi(q, d)$ tries to distinguish between relevant documents and non-relevant documents. Similar to the objective function of GAN, the overall objective is formulated as follows:

$$J^{G^*,D^*} = \min_\theta \max_\phi \sum_{n=1}^{N} (\mathbb{E}_{d\sim p_{true}(d|q_n,r)}[logD(d|q_n)] + \mathbb{E}_{d\sim p_\theta(d|q_n,r)}[log(1 - D(d|q_n))]) \tag{15}$$

where $D(d|q_n) = \sigma(f_\phi(q, d))$, σ represents the sigmoid function, θ and ϕ are the parameters for generative and discriminative retrieval respectively. Parameter θ and ϕ can be learned alternately with gradient descent.

The above objective equation is constructed for pointwise relevance estimation. In some specific tasks, it should be in pairwise paradigm to generate higher quality ranking lists. Here, suppose $p_\theta(d|q_n, r)$ is given by a softmax function:

$$p_\theta(d_i|q, r) = \frac{exp(g_\theta(q, d_i))}{\sum_{d_j} exp(g_\theta(q, d_j))} \tag{16}$$

$g_\theta(q, d)$ is the chance of document d being generated from query q. In real-word retrieval system, both $g_\theta(q, d)$ and $f_\phi(q, d)$ are task-specific. They can either have the same or different formulations. The authors modelled them with the same function for convenience, and define them as: $g_\theta(q, d) = s_\theta(q, d)$ and $f_\phi(q, d) = s_\phi(q, d)$. In the item recommendation scenario, the authors adopted the matrix factorization to formulate $s(\cdot)$. It can be substituted with other advanced models such as factorization machine or neural network.

He et al. [52] proposed an adversarial personalized ranking approach which enhances the Bayesian personalized ranking with adversarial training. It plays a minimax game between the original BPR objective and the adversary which add noises or permutations to maximize the BPR loss. Cai et al. [9] proposed a GAN based representation learning approach for heterogeneous bibliographic network, which can effectively address the personalized citation recommendation task. Wang et al. [164] proposed using GAN to generate negative samples for the memory network based streaming recommender. Experiments show that the proposed GAN based sampler could significantly improve the performance.

3.11 Deep Hybrid Models for Recommendation

With the good flexibility of deep neural networks, many neural building blocks can be intergrated to formalize more powerful and expressive models. Despite the abundant possible ways of combination, we suggest that the hybrid model should be reasonably and carefully designed for the specific tasks. Here, we summarize the existing models that has been proven to be effective in some application fields.

CNNs and Autoencoder. Collaborative Knowledge Based Embedding (CKE) [192] combines CNNs with autoencoder for images feature extraction. CKE can be viewed as a further step of CDL. CDL only considers item text information (e.g. abstracts of articles and plots of movies), while CKE leverages structural content, textual content and visual content with different embedding techniques. Structural information includes the attributes of items and the relationships among items and users. CKE adopts the TransR [96], a heterogeneous network embedding method, for interpreting structural information. Similarly, CKE employs SDAE to learn feature representations from textual information. As for visual information, CKE adopts a stacked convolutional auto-encoders (SCAE). SCAE makes efficient use of convolution by replacing the fully-connected layers of SDAE with convolutional layers. The recommendation process is done in a probabilistic form similar to CDL.

CNNs and RNNs. Lee et al. [82] proposed a deep hybrid model with RNNs and CNNs for quotes recommendation. Quote recommendation is viewed as a task of generating a ranked list of quotes given the query texts or dialogues (each dialogue contains a sequence of tweets). It applies CNN sto learn significant local semantics from tweets and maps them to a distributional vectors. These distributional vectors are further processed by LSTM to compute the relevance of target quotes to the given tweet dialogues. The overall architecture is shown in Figure 12(a).

Zhang et al. [193] proposed a CNNs and RNNs based hybrid model for hashtag recommendation. Given a tweet with corresponding images, the authors utilized CNNs to extract features from images and LSTM to learn text features from tweets. Meanwhile, the authors proposed a co-attention mechanism to model the correlation influences and balance the contribution of texts and images.

Ebesu et al. [38] presented a neural citation network which integrates CNNs with RNNs in a encoder-decoder framework for citation recommendation. In this model, CNNs act as the encoder that captures the long-term dependencies from citation context. The RNNs work as a decoder which learns the probability of a word in the cited paper's title given all previous words together with representations attained by CNNs.

Chen et al. [17] proposed an intergrated framework with CNNs and RNNs for personalized key frame (in videos) recommendation, in which CNNs are used to learn feature representations from key frame images and RNNs are used to process the textual features.

RNNs and Autoencoder. The former mentioned collaborative deep learning model is lack of robustness and incapable of modelling the sequences of text information. Wang et al. [160] further exploited integrating RNNs and denoising autoencoder to overcome this limitations. The authors first designed a generalization of RNNs named robust recurrent network. Based on the robust recurrent network, the authors proposed the hierarchical

Bayesian recommendation model called CRAE. CRAE also consists of encoding and decoding parts, but it replaces feedforward neural layers with RNNs, which enables CRAE to capture the sequential information of item content information. Furthermore, the authors designed a wildcard denoising and a beta-pooling technique to prevent the model from overfitting.

RNNs with DRL. Wang et al. [163] proposed combining supervised deep reinforcement learning wth RNNs for treatment recommendation. The framework can learn the prescription policy from the indicator signal and evaluation signal. Experiments demonstrate that this system could infer and discover the optimal treatments automatically. We believe that this a valuable topic and benefits the social good.

4 FUTURE RESEARCH DIRECTIONS AND OPEN ISSUES

Whilst existing works have established a solid foundation for deep recommender systems research, this section outlines several promising prospective research directions. We also elaborate on several open issues, which we believe is critical to the present state of the field.

4.1 Joint Representation Learning from User and Item Content Information

Making accurate recommendations requires deep understanding of item characteristics and user's actual demands and preferences [1, 85]. Naturally, this can be achieved by exploiting the abundant auxiliary information. For example, context information tailors services and products according to user's circumstances and surroundings [151], and mitigate cold start influence; implicit feedback indicates users' implicit intention and is easier to collect while gathering explicit feedback is a resource-demanding task. Although existing works have investigated the efficacy of deep learning model in mining user and item profiles [92, 196], implicit feedback [50, 188, 196, 203], contextual information [38, 75, 118, 149, 151], and review texts [87, 127, 174, 202] for recommendation, they do not utilize these various side information in a comprehensive manner and take the full advantages of the available data. Moreover, there are few works investigating users' footprints (e.g. Tweets or Facebook posts) from social media [61] and physical world (e.g. Internet of things) [186]. One can infer user's temporal interests or intentions from these side data resources while deep learning method is a desirable and powerful tool for integrating these additional information. The capability of deep learning in processing heterogeneous data sources also brings more opportunities in recommending diverse items with unstructured data such as textual, visual, audio and video features.

Additionally, feature engineering has not been fully studied in the recommendation research community, but it is essential and widely employed in industrial applications [20, 27]. However, most of the existing models require manually crafted and selected features, which is time-consuming and tedious. Deep neural network is a promising tool for automatic feature crafting by reducing manual intervention [129]. There is also an added advantage of representation learning from free texts, images or data that exists in the 'wild' without having to design intricate feature engineering pipelines. More intensive studies on deep feature engineering specific for recommender systems are expected to save human efforts as well as improve recommendation quality.

An interesting forward looking research problem is how to design neural architectures that best exploits the availability of other modes of data. One recent work potentially paving the way towards models of this nature is the Joint Representation Learning framework [197]. Learning joint (possibly multi-modal representations) of user and items will likely become a next emerging trend in recommender systems research. To this end, a deep learning taking on this aspect would be how to design better inductive biases (hybrid neural architectures) in an end-to-end fashion. For example, reasoning over different modalities (text, images, interaction) data for better recommendation performance.

4.2 Explainable Recommendation with Deep Learning

A common interpretation is that deep neural networks are highly non-interpretable. As such, making explainable recommendations seem to be an uphill task. Along the same vein, it would be also natural to assume that big, complex neural models are just fitting the data with any *true* understanding (see subsequent section on machine reasoning for recommendation). This is precisely why this direction is both exciting and also crucial. There are mainly two ways that explainable deep learning is important. The first, is to make explainable predictions to users, allowing them to understand the factors behind the network's recommendations (i.e., why was this item/service recommended?) [126, 178]. The second track is mainly focused on explain-ability to the practitioner, probing weights and activations to understand more about the model [145].

As of today, attentional models [126, 146, 178] have more or less eased the non-interpretable concerns of neural models. If anything, attention models have instead led to greater extents of interpretability since the attention weights not only give insights about the inner workings of the model but are also able to provide explainable results to users. While this has been an existing direction of research 'pre deep learning', attentional models are not only capable of enhancing performance but enjoys greater explainability. This further motivates the usage of deep learning for recommendation.

Notably, it is both intuitive and natural that a model's explainabiity and interpretability strongly relies on the application domain and usage of content information. For example [126, 146] mainly use reviews as a medium of interpretability (which reviews led to making which predictions). Many other mediums/modalities can be considered, such as image [18].

To this end, a promising direction and next step would be to design *better* attentional mechanisms, possibly to the level of providing conversational or generative explanations (along the likes of [87]). Given that models are already capable of highlighting what contributes to the decision, we believe that this is the next frontier.

4.3 Going Deeper for Recommendation

From former studies [53, 53, 177, 195], we found that the performance of most neural CF models plateaus at three to four layers. Going deeper has shown promising performance over shallow networks in many tasks [48, 64], nonetheless, going deeper in the context of deep neural network based RS remains largely unclear. If going deeper give favorable results, how do we train the deep architecture? If not, what is the reason behind this? A possibility is to look into auxiliary losses at different layers in similar spirit to [147] albeit hierarchically instead of sequentially. Another possibility is to vary layer-wise learning rates for each layer of the deep network or apply some residual strategies.

4.4 Machine Reasoning for Recommendation

There have been numerous recent advances in *machine reasoning* in deep learning, often involving reasoning over natural language or visual input [67, 124, 181]. We believe that tasks like machine reading, reasoning, question answering or even visual reasoning will have big impacts on the field of recommender systems. These tasks are often glazed over, given that they seem completely arbitrary and irrelevant with respect to recommender systems. However, it is imperative that recommendater systems often requires reasoning over a single (or multiple) modalities (reviews, text, images, meta-data) which would eventually require borrowing (and adapting) techniques from these related fields. Fundamentally, recommendation and reasoning (e.g., question answering) are highly related in the sense that they are both information retrieval problems.

The single most impactful architectural innovation with neural architectures that are capable of machine reasoning is the key idea of attention [155, 181]. Notably, this key intuition have already (and very recently) demonstrated effectiveness on several recommender problems. Tay et al. [146] proposed an co-attentive architecture for *reasoning over reviews*, and showed that different recommendation domains have different 'evidence aggregation’ patterns. For interaction-only recommendation, similar reasoning architectures have utilized similar attentive mechanisms for reasoning over meta-paths [62]. To this end, a next frontier for recommender systems is possibly to adapt to situations that require multi-step inference and reasoning. A simple example would to reason over a user’s social profile, purchases etc., reasoning over multiple modalities to recommend a product. All in all, we can expect that reasoning architectures to start to take the foreground in recommender system research.

4.5 Cross Domain Recommendation with Deep Neural Networks

Nowadays, many large companies offer diversified products or services to customers. For example, Google provides us with web searches, mobile applications and news services; We can buy books, electronics and clothes from Amazon. Single domain recommender system only focuses on one domain while ignores the user interests on other domains, which also exacerbates sparsity and cold start problems [74]. Cross domain recommender system, which assists target domain recommendation with the knowledge learned from source domains, provides a desirable solution for these problems. One of the most widely studied topics in cross domain recommendation is transfer learning which aims to improve learning tasks in one domain by using knowledge transferred from other domains [40, 115]. Deep learning is well suited to transfer learning as it learn high-level abstractions that disentangle the variation of different domains. Several existing works [39, 92] indicate the efficacy of deep learning in catching the generalizations and differences across different domains and generating better recommendations on cross-domain platforms. Therefore, it is a promising but largely under-explored area where mores studies are expected.

4.6 Deep Multi-Task Learning for Recommendation

Multi-task learning has led to successes in many deep learning tasks, from computer vision to natural language processing [26, 31]. Among the reviewed studies, several works [5, 73, 87, 187] also applied multi-task learning to recommender system in a deep neural framework and achieved some improvements over single task learning. The advantages of applying deep neural network based multi-task learning are three-fold: (1) learning several tasks at a time can prevent overfitting by generalizing the shared hidden representations; (2) auxiliary task provides interpretable output for explaining the recommendation; (3) multi-task provides an implicit data augmentation for alleviating the sparsity problem. Multitask can be utilized in traditional recommender system [111], while deep learning enables them to be integrated in a tighter fashion. Apart from introducing side tasks, we can also deploy the multitask learning for cross domain recommendation with each specific task generating recommendation for each domain.

4.7 Scalability of Deep Neural Networks for Recommendation

The increasing data volumes in the big data era poses challenges to real-world applications. Consequently, scalability is critical to the usefulness of recommendation models in real-world systems, and the time complexity will also be a principal consideration for choosing models. Fortunately, deep learning has demonstrated to be very effective and promising in big data analytics [109] especially with the increase of GPU computation power. However, more future works should be studied on how to recommend efficiently by exploring the following problems: (1) incremental learning for non-stationary and streaming data such as large volume of incoming users and items; (2) computation efficiency for high-dimensional tensors and multimedia data sources; (3) balancing of the model complexity and scalability with the exponential growth of parameters. A promising area of research in this area involves knowledge distillation which have been explored in [144] for learning small/compact models for inference in recommender systems. The key idea is to train a smaller student model that absorbs knowledge from the large teacher model. Given that inference time is crucial for real time applications at a million/billion user scale, we believe that this is another promising direction which warrants further investigation. Another promising direction involves compression techniques [128]. The high-dimensional input data can be compressed to compact embedding to reduce the space and computation time during model learning.

4.8 The Field Needs Better, More Unified and Harder Evaluation

Each time a new model is proposed, it is expected that the publication offers evaluation and comparisons against several baselines. The selection of baselines and datasets on most papers are seemingly arbitrary and authors generally have free reign over the choices of datasets/baselines. There are several issues with this.

Firstly, this creates an inconsistent reporting of scores, with each author reporting their own assortment of results. Till this day, there is seemingly on consensus on a general ranking of models (Notably, we acknowledge that the *no free lunch theorem* exists). Occasionally, we find that results can be conflicting and relative positions change very frequently. For example, the scores of NCF in [201] is relatively ranked very low as compared to the original paper that proposed the model [53]. This makes the relative benchmark of new neural models extremely challenging. The question is how do we solve this? Looking into neighbouring fields (computer vision or natural language processing), this is indeed perplexing. Why is there no MNIST, ImageNet or SQuAD for recommender systems? As such, we believe that a suite of standardized evaluation datasets should be proposed.

We also note that datasets such as MovieLens are commonly used by many practioners in evaluating their models. However, the splits are often arbitrary (randomized). The second problem is that there is no control over the evaluation procedure. To this end, we urge the recommender systems community to follow the CV/NLP communities and establish a hidden/blinded test set in which prediction results can be only submitted via a web interface (such as Kaggle).

Finally, a third recurring problem is that there is no control over the difficulty of test samples in recommender system result. Is splitting by time the best? How do we know if test samples are either too trivial or impossible to infer? Without designing proper test sets, we argue that it is in fact hard to estimate and measure progress of the field. To this end, we believe that the field of recommender systems have a lot to learn from computer vision or NLP communities.

5 CONCLUSION

In this article, we provided an extensive review of the most notable works to date on deep learning based recommender systems. We proposed a classification scheme for organizing and clustering existing publications, and highlighted a bunch of influential research prototypes. We also discussed the advantages/disadvantages of using deep learning techniques for recommendation tasks. Additionally, we detail some of the most pressing open problems and promising future extensions. Both deep learning and recommender systems are ongoing hot research topics in the recent decades. There are a large number of new developing techniques and emerging models each year. We hope this survey can provide readers with a comprehensive understanding towards the key aspects of this field, clarify the most notable advancements and shed some light on future studies.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17, 6 (2005), 734–749.

[2] Taleb Alashkar, Songyao Jiang, Shuyang Wang, and Yun Fu. 2017. Examples-Rules Guided Deep Neural Network for Makeup Recommendation. In *AAAI*. 941–947.

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2014. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755* (2014).

[4] Bing Bai, Yushun Fan, Wei Tan, and Jia Zhang. 2017. DLTSR: A Deep Learning Framework for Recommendation of Long-tail Web Services. *IEEE Transactions on Services Computing* (2017).

[5] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 107–114.

[6] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[7] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. 2017. Deep Learning Methods on Recommender System: A Survey of State-of-the-art. *International Journal of Computer Applications* 162, 10 (Mar 2017).

[8] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

[9] Xiaoyan Cai, Junwei Han, and Libin Yang. 2018. Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation. In *AAAI*.

[10] S. Cao, N. Yang, and Z. Liu. 2017. Online news recommender based on stacked auto-encoder. In *ICIS*. 721–726.

[11] Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Recsys*. 288–296.

[12] [illegible] Chen, Xiangwu Meng, Zhenghua Xu, and Thomas Lukasiewicz. 2017. Location-Aware Personalized News Recommendation with Deep Semantic Analysis. *IEEE Access* 5 (2017), 1624–1638.

[13] [illegible] Chen, Peilin Zhao, Longfei Li, Jun Zhou, Xiaolong Li, and Minghui Qiu. 2017. Locally Connected Deep Learning Framework for Industrial-scale Recommender Systems. In *WWW*.

[14] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. (2017).

[15] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683* (2012).

[16] Shi-Yong Chen, Yang Yu, Qing Da, Jun Tan, Hai-Kuan Huang, and Hai-Hong Tang. 2018. Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *SIGKDD*. 1187–1196.

[17] Xu Chen, Yongfeng Zhang, Qingyao Ai, Hongteng Xu, Junchi Yan, and Zheng Qin. 2017. Personalized Key Frame Recommendation. In *SIGIR*.

[18] Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Visually Explainable Recommendation. *arXiv preprint arXiv:1801.10288* (2018).

[19] Yifan Chen and Maarten de Rijke. 2018. A Collective Variational Autoencoder for Top-N Recommendation with Side Information. *arXiv preprint arXiv:1807.05730* (2018).

[20] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In *Recsys*. 7–10.

[21] Sungwoon Choi, Heonseok Ha, Uiwon Hwang, Chanju Kim, Jung-Woo Ha, and Sungroh Yoon. 2018. Reinforcement Learning based Recommender System using Biclustering Technique. *arXiv preprint arXiv:1801.05532* (2018).

[22] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: first results. *arXiv preprint arXiv:1412.1602* (2014).

[23] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*. 577–585.

[24] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. 2018. Q&R: A Two-Stage Approach toward Interactive Recommendation. In *SIGKDD*. 139–148.

[25] Wei-Ta Chu and Ya-Lun Tsai. 2017. A hybrid recommendation system considering visual information for predicting favorite restaurants. *WWWJ* (2017), 1–19.

[26] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. 160–167.

[27] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Recsys*. 191–198.

[28] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep coevolutionary network: Embedding user and item features for recommendation. arXiv preprint. *arXiv preprint arXiv:1609.03675* (2016).

[29] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In *Recsys*. 29–34.

[30] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube Video Recommendation System. In *Recsys*.

[31] Li Deng, Dong Yu, and others. 2014. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing* 7, 3–4 (2014), 197–387.

[32] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. 2017. On deep learning for trust-aware recommendations in social networks. *IEEE transactions on neural networks and learning systems* 28, 5 (2017), 1164–1177.

[33] Robin Devooght and Hugues Bersini. 2016. Collaborative filtering with recurrent neural networks. *arXiv preprint arXiv:1608.07400* (2016).

[34] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. 2017. A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems. In *AAAI*. 1309–1315.

[35] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Recsys*. 152–160.

[36] Chao Du, Chongxuan Li, Yin Zheng, Jun Zhu, and Bo Zhang. 2016. Collaborative Filtering with User-Item Co-Autoregressive Models. *arXiv preprint arXiv:1612.07146* (2016).

[37] Gintare Karolina Dziugaite and Daniel M Roy. 2015. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443* (2015).

[38] Travis Ebesu and Yi Fang. 2017. Neural Citation Network for Context-Aware Citation Recommendation. (2017).

[39] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*. 278–288.

[40] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*. 24.

[41] Jianfeng Gao, Li Deng, Michael Gamon, Xiaodong He, and Patrick Pantel. 2014. Modeling interestingness with deep neural networks. (June 13 2014). US Patent App. 14/304,863.

[42] Kostadin Georgiev and Preslav Nakov. 2013. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *ICML*. 1148–1156.

[43] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *TMIS* 6, 4 (2016), 13.

[44] Yuyun Gong and Qi Zhang. 2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In *IJCAI*. 2782–2788.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*. 2672–2680.

[47] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 2782–2788.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[49] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*. 507–517.

[50] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*. 144–150.

[51] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer Product-based Neural Collaborative Filtering. (2018).

[52] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *SIGIR*. 355–364.

[53] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[54] Xiangnan He and Chua Tat-Seng. 2017. Neural Factorization Machines for Sparse Predictive Analytics. (2017).

[55] Balázs Hidasi and Alexandros Karatzoglou. 2017. Recurrent neural networks with top-k gains for session-based recommendations. *arXiv preprint arXiv:1706.03847* (2017).

[56] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations* (2015).

[57] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Recsys*. 241–248.

[58] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks* 4, 2 (1991), 251–257.

[59] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[60] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *WWW*. 193–201.

[61] Cheng-Kang Hsieh, Longqi Yang, Honghao Wei, Mor Naaman, and Deborah Estrin. 2016. Immersive recommendation: News and event recommendations using personal digital traces. In *WWW*. 51–62.

[62] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging Meta-path based Context for Top-N Recommendation with A Neural Co-Attention Model. In *SIGKDD*. 1531–1540.

[63] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*.

[64] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks.. In *CVPR*, Vol. 1. 3.

[65] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. 2333–2338.

[66] Wenyi Huang, Zhaohui Wu, Liang Chen, Prasenjit Mitra, and C Lee Giles. 2015. A Neural Probabilistic Model for Context Based Citation Recommendation. In *AAAI*. 2404–2410.
[67] Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067* (2018).
[68] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Recsys*.
[69] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction.*
[70] Yogesh Jhamb, Travis Ebesu, and Yi Fang. 2018. Attentive Contextual Denoising Autoencoder for Recommendation. (2018).
[71] X. Jia, X. Li, K. Li, V. Gopalakrishnan, G. Xun, and A. Zhang. 2016. Collaborative restricted Boltzmann machine for social event recommendation. In *ASONAM*. 402–405.
[72] Xiaowei Jia, Aosen Wang, Xiaoyi Li, Guangxu Xun, Wenyao Xu, and Aidong Zhang. 2015. Multi-modal learning for video recommendation based on mobile application usage. In *2015 IEEE International Conference on Big Data (Big Data)*. 837–842.
[73] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *WSDM*. 515–524.
[74] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross Domain Recommender Systems: A Systematic Literature Review. *ACM Comput. Surv.* 50, 3 (June 2017).
[75] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Recsys*. 233–240.
[76] Donghyun Kim, Chanyoung Park, Jinoh Oh, and Hwanjo Yu. 2017. Deep Hybrid Recommender Systems via Exploiting Document Context and Statistics of Items. *Information Sciences* (2017).
[77] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[78] Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. 2016. Collaborative recurrent neural networks for dynamic recommender systems. In *Asian Conference on Machine Learning*. 366–381.
[79] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*. 426–434.
[80] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.
[81] Hugo Larochelle and Iain Murray. 2011. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 29–37.
[82] Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote Recommendation in Dialogue using Deep Neural Network. In *SIGIR*. 957–960.
[83] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Paul Natsev. 2018. Collaborative Deep Metric Learning for Video Understanding. (2018).
[84] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. 2016. Comparative Deep Learning of Hybrid Representations for Image Recommendations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2545–2553.
[85] Jure Leskovec. 2015. New Directions in Recommender Systems. In *WSDM*.
[86] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.
[87] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. (2017).
[88] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *CIKM*. 811–820.
[89] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *SIGKDD*.
[90] Yang Li, Ting Liu, Jing Jiang, and Liang Zhang. 2016. Hashtag recommendation with topical attention-based LSTM. In *COLING*.
[91] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from History and Present: Next-item Recommendation via Discriminatively Exploiting User Behaviors. In *SIGKDD*. 1734–1743.
[92] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. In *WWW*. 817–818.
[93] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. *arXiv preprint arXiv:1803.05170* (2018).
[94] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. *arXiv preprint arXiv:1802.05814* (2018).
[95] Dawen Liang, Minshu Zhan, and Daniel PW Ellis. 2015. Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks.. In *ISMIR*. 295–301.
[96] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. 2181–2187.
[97] Juntao Liu and Caihua Wu. 2017. *Deep Learning Based Recommendation: A Survey.*
[98] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning User Preferences for Visual Recommendation. (2017).

752-753 2 notes:
754-756 3 notes:
757-758 2 notes:
759-761 3 notes:
762-764 3 notes:
765-766 2 notes:
767-770 4 notes:
771-773 3 notes:
774-775 2 notes:
776-778 3 notes:
779-780 2 notes:
781
fan zr

[99] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *SIGKDD*. 1831–1839.

[100] Xiaomeng Liu, Yuanxin Ouyang, Wenge Rong, and Zhang Xiong. 2015. Item Category Aware Conditional Restricted Boltzmann Machine Based Recommendation. In *International Conference on Neural Information Processing*. 609–616.

[101] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture. In *Recsys*. 147–151.

[102] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture. In *Recsys (RecSys '17)*.

[103] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *CoRR* abs/1803.09587 (2018).

[104] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[105] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. 43–52.

[106] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and others. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[107] Isshu Munemasa, Yuta Tomomatsu, Kunioki Hayashi, and Tomohiro Takagi. 2018. Deep Reinforcement Learning for Recommender Systems. (2018).

[108] Cataldo Musto, Claudio Greco, Alessandro Suglia, and Giovanni Semeraro. 2016. Ask Me Any Rating: A Content-based Recommender System based on Recurrent Neural Networks. In *IIR*.

[109] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, 1 (2015), 1.

[110] Hanh T. H. Nguyen, Martin Wistuba, Josif Grabocka, Lucas Rego Drumond, and Lars Schmidt-Thieme. 2017. *Personalized Deep Learning for Tag Recommendation.*

[111] Xia Ning and George Karypis. 2010. Multi-task learning for recommender system. In *Proceedings of 2nd Asian Conference on Machine Learning*. 269–284.

[112] Wei Niu, James Caverlee, and Haokai Lu. 2018. Neural Personalized Ranking for Image Recommendation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 423–431.

[113] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based News Recommendation for Millions of Users. In *SIGKDD*.

[114] Yuanxin Ouyang, Wenqi Liu, Wenge Rong, and Zhang Xiong. 2014. Autoencoder-based collaborative filtering. In *International Conference on Neural Information Processing*. 284–291.

[115] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *AAAI*, Vol. 10. 230–235.

[116] Yiteng Pana, Fazhi Hea, and Haiping Yua. 2017. Trust-aware Collaborative Denoising Auto-Encoder for Top-N Recommendation. *arXiv preprint arXiv:1703.01760* (2017).

[117] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Recsys*. 130–137.

[118] Yogesh Singh Rawat and Mohan S Kankanhalli. 2016. ConTagNet: exploiting user context for image tag recommendation. In *Proceedings of the 2016 ACM on Multimedia Conference*. 1102–1106.

[119] S. Rendle. 2010. Factorization Machines. In *2010 IEEE International Conference on Data Mining*.

[120] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. 452–461.

[121] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender systems handbook*. 1–34.

[122] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*. 833–840.

[123] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *ICML*. 791–798.

[124] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *NIPS*. 4967–4976.

[125] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*. 111–112.

[126] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Recsys*. 297–305.

[127] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Representation Learning of Users and Items for Review Rating Prediction Using Attention-based Convolutional Neural Network. In *MLRec*.

[128] Joan Serrà and Alexandros Karatzoglou. 2017. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In *Recsys*. 279–287.

[129] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep Crossing: Web-scale modeling without manually crafted combinatorial features. In *SIGKDD*. 255–262.

[130] Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Automatic Recommendation Technology for Learning Resources with Convolutional Neural Network. In *International Symposium on Educational Technology*. 30–34.

[131] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 3.

[132] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. (2017).

[133] Harold Soh, Scott Sanner, Madeleine White, and Greg Jamieson. 2017. Deep Sequential Recommendation for Personalized Adaptive User Interfaces. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. 589–593.

[134] Bo Song, Xin Yang, Yi Cao, and Congfu Xu. 2018. Neural Collaborative Ranking. *arXiv preprint arXiv:1808.04957* (2018).

[135] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *SIGIR*. 909–912.

[136] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid Recommender System based on Autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 11–16.

[137] Florian Strub and Jeremie Mary. 2015. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In *NIPS Workshop*.

[138] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[139] Alessandro Suglia, Claudio Greco, Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2017. A Deep Architecture for Content-based Recommendations Exploiting Recurrent Neural Networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 202–211.

[140] Yosuke Suzuki and Tomonobu Ozaki. 2017. Stacked Denoising Autoencoder-Based Deep Collaborative Filtering Using the Change of Similarity. In *WAINA*. 498–502.

[141] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A Neural Network Approach to Quote Recommendation in Writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 65–74.

[142] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Recsys*. 17–22.

[143] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[144] Jiaxi Tang and Ke Wang. 2018. Ranking Distillation: Learning Compact Ranking Models With High Performance for Recommender System. In *SIGKDD*.

[145] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In *WWW*.

[146] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *SIGKDD*.

[147] Trieu H Trinh, Andrew M Dai, Thang Luong, and Quoc V Le. 2018. Learning longer-term dependencies in rnns with auxiliary losses. *arXiv preprint arXiv:1803.00144* (2018).

[148] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D Convolutional Networks for Session-based Recommendation with Content Features. In *Recsys*. 138–146.

[149] Bartłomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Recsys*.

[150] Moshe Unger. 2015. Latent Context-Aware Recommender Systems. In *Recsys*. 383–386.

[151] Moshe Unger, Ariel Bar, Bracha Shapira, and Lior Rokach. 2016. Towards latent context-aware recommendation systems. *Knowledge-Based Systems* 104 (2016), 165–178.

[152] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. 2016. Neural autoregressive distribution estimation. *Journal of Machine Learning Research* 17, 205 (2016), 1–37.

[153] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *NIPS*. 2643–2651.

[154] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A Meta-Learning Perspective on Cold-Start Recommendations for Items. In *Advances in Neural Information Processing Systems*. 6904–6914.

[155] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[156] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. In *Advances in Neural Information Processing Systems*. 4957–4966.

[157] Jeroen B. P. Vuurens, Martha Larson, and Arjen P. de Vries. 2016. Exploring Deep Space: Learning Personalized Ranking in a Semantic Space. In *Recsys*.

[158] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2015. Relational Stacked Denoising Autoencoder for Tag Recommendation.. In *AAAI*. 3052–3058.

[159] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *SIGKDD*. 1235–1244.

[160] Hao Wang, SHI Xingjian, and Dit-Yan Yeung. 2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In *NIPS*. 415–423.

[161] Hao Wang and Dit-Yan Yeung. 2016. Towards Bayesian deep learning: A framework and some existing methods. *TKDE* 28, 12 (2016), 3395–3408.

[162] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. (2017).

[163] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. 2018. Supervised Reinforcement Learning with Recurrent Neural Network for Dynamic Treatment Recommendation. In *SIGKDD*. 2447–2456.

[164] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural Memory Streaming Recommender Networks with Adversarial Training. In *SIGKDD*.

[165] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What Your Images Reveal: Exploiting Visual Contents for Point-of-Interest Recommendation. In *WWW*.

[166] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. (2017).

[167] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *MM*. 627–636.

[168] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. 2014. Exploration in interactive personalized music recommendation: a reinforcement learning approach. *TOMM* 11, 1 (2014), 7.

[169] Xuejian Wang, Lantao Yu, Kan Ren, Guangyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic Attention Deep Model for Article Recommendation by Learning Human Editorsfi Demonstration. In *SIGKDD*.

[170] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2016. Collaborative filtering and deep learning based hybrid recommendation for cold start problem. IEEE, 874–877.

[171] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications* 69 (2017), 29–39.

[172] Jiqing Wen, Xiaopeng Li, James She, Soochang Park, and Ming Cheung. 2016. Visual background recommendation for dance performances using dancer-shared images. 521–527.

[173] Caihua Wu, Junwei Wang, Juntao Liu, and Wenyu Liu. 2016. Recurrent neural network based recommendation for time heterogeneous feedback. *Knowledge-Based Systems* 109 (2016), 90–103.

[174] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, and Alexander J Smola. 2016. Joint Training of Ratings and Reviews with Recurrent Recommender Networks. (2016).

[175] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. 495–503.

[176] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. 2016. Personal recommendation using deep recurrent neural networks in NetEase. In *ICDE*. 1218–1229.

[177] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*. 153–162.

[178] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[179] Ruobing Xie, Zhiyuan Liu, Rui Yan, and Maosong Sun. 2016. Neural Emoji Recommendation in Dialogue Systems. *arXiv preprint arXiv:1612.04609* (2016).

[180] Weizhu Xie, Yuanxin Ouyang, Jingshuai Ouyang, Wenge Rong, and Zhang Xiong. 2016. User Occupation Aware Conditional Restricted Boltzmann Machine Based Recommendation. 454–461.

[181] Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604* (2016).

[182] Zhenghua Xu, Cheng Chen, Thomas Lukasiewicz, Yishu Miao, and Xiangwu Meng. 2016. Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. In *CIKM*. 1921–1924.

[183] Zhenghua Xu, Thomas Lukasiewicz, Cheng Chen, Yishu Miao, and Xiangwu Meng. 2017. Tag-aware personalized recommendation using a hybrid deep model. (2017).

[184] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*. 3203–3209.

[185] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In *SIGKDD*.

[186] Lina Yao, Quan Z Sheng, Anne HH Ngu, and Xue Li. 2016. Things of interest recommendation by leveraging heterogeneous relations in the internet of things. *ACM Transactions on Internet Technology (TOIT)* 16, 2 (2016), 9.

[187] Baolin Yi, Xiaoxuan Shen, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Expanded autoencoder recommendation framework and its application in movie recommendation. In *SKIMA*. 298–303.

[188] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. 2016. Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback. In *PAKDD*. 555–567.

[189] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *IJCAI*.

[190] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *arXiv preprint arXiv:1806.01973* (2018).

[191] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *WWW*. 649–658.

[192] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*. 353–362.

[193] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In *IJCAI*.

[194] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. 2018. Next Item Recommendation with Self-Attention. *arXiv preprint arXiv:1808.06414* (2018).

[195] Shuai Zhang, Lina Yao, Aixin Sun, Sen Wang, Guodong Long, and Manqing Dong. 2018. NeuRec: On Nonlinear Transformation for Personalized Ranking. *arXiv preprint arXiv:1805.03002* (2018).

[196] Shuai Zhang, Lina Yao, and Xiwei Xu. 2017. AutoSVD++: An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. (2017).

[197] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *CIKM*. 1449–1458.

[198] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep Reinforcement Learning for Page-wise Recommendations. *arXiv preprint arXiv:1805.02343* (2018).

[199] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. *arXiv preprint arXiv:1802.06501* (2018).

[200] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In *WWW*. 167–176.

[201] Lei Zheng, Chun-Ta Lu, Lifang He, Sihong Xie, Vahid Noroozi, He Huang, and Philip S Yu. 2018. MARS: Memory Attention-Aware Recommender System. *arXiv preprint arXiv:1805.07037* (2018).

[202] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*.

[203] Yin Zheng, Cailiang Liu, Bangsheng Tang, and Hanning Zhou. 2016. Neural Autoregressive Collaborative Filtering for Implicit Feedback. In *Recsys*.

[204] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A Neural Autoregressive Approach to Collaborative Filtering. In *ICML*.

[205] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2017. ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation. *arXiv preprint arXiv:1711.06632* (2017).

[206] Jiang Zhou, Cathal Gurrin, and Rami Albatal. 2016. Applying visual user interest profiles for recommendation & personalisation. (2016).

[207] Fuzhen Zhuang, Dan Luo, Nicholas Jing Yuan, Xing Xie, and Qing He. 2017. Representation Learning with Pair-wise Constraints for Collaborative Ranking. In *WSDM*. 567–575.

[208] Fuzhen Zhuang, Zhiqiang Zhang, Mingda Qian, Chuan Shi, Xing Xie, and Qing He. 2017. Representation learning via Dual-Autoencoder for recommendation. *Neural Networks* 90 (2017), 83–89.

[209] Yi Zuo, Jiulin Zeng, Maoguo Gong, and Licheng Jiao. 2016. Tag-aware recommender systems based on deep neural networks. *Neurocomputing* 204 (2016), 51–60.

Deep Learning based Recommender System: A Survey and New Perspectives

Zhang, Shuai; Yao, Lina; Sun, Aixin; Tay, Yi

01 fan zr Page 1

21/9/2019 2:08

非常全面详尽的推荐系统领域综述，值得反复阅读
对近几年深度学习在推荐系统领域的发展做了详细的整理和总结，基于CNN，RNN，RBM，Attention等模型的推荐技术是需要逐一深入了解的方向

02 fan zr Page 1

10/9/2019 16:21

03 fan zr Page 1

10/9/2019 16:24

69： Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. Recommender systems: an introduction.

121：Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In Recommender systems handbook. 1–34.

04 fan zr Page 1

11/9/2019 2:16

05 fan zr Page 2

11/9/2019 2:17

根据输入数据类型可以把推荐系统分为三类：协同过滤，基于内容，混合方法

06 fan zr Page 2

11/9/2019 2:16

07 fan zr Page 2

11/9/2019 2:16

08 fan zr Page 2

11/9/2019 2:16

09 fan zr Page 2

11/9/2019 2:17

10 fan zr Page 2

11/9/2019 2:21

2016. Deep neural networks for youtube recommendations. In Recsys.

11 fan zr Page 2

11/9/2019 7:19

12 fan zr Page 2

11/9/2019 14:01

13 fan zr Page 2

11/9/2019 13:58

2016. Wide & deep learning for recommender systems. In Recsys
2010. The YouTube Video Recommendation System. In Recsys.
The netflix recommender system: Algorithms, business value, and innovation. TMIS 6, 4 (2016), 13

2017. Embedding-based News Recommendation for Millions of Users. In SIGKDD.
[114]

14 fan zr Page 2

11/9/2019 14:02

15 fan zr Page 2

12/9/2019 1:57

2009. A survey of collaborative filtering techniques. Advances

16 fan zr Page 2

12/9/2019 1:58

2002. Hybrid recommender systems: Survey and experiments

17 fan zr Page 2

12/9/2019 1:59

2012. Cross-domain recommender systems: A survey of the state of the art

18 fan zr Page 2

12/9/2019 1:59

2017. Cross Domain Recommender Systems: A Systematic Literature Review.

19 fan zr Page 3

12/9/2019 2:10

2017. Deep Learning Methods on Recommender System: A Survey of State-of-the-art.

2017. Deep Learning Based Recommendation: A Survey.

20 fan zr Page 3

12/9/2019 2:11

2007. Restricted Boltzmann machines for collaborative filtering. In ICML.
2013. Deep content-based music recommendation. In NIPS
2015. Collaborative deep learning for recommender systems. In SIGKDD

21 fan zr Page 3

12/9/2019 2:13

22 fan zr Page 3

12/9/2019 3:54

23 fan zr Page 3

12/9/2019 4:00

24 fan zr Page 3

12/9/2019 4:07

论文来源：
谷歌学术，WebOfScience， NIPS，ICML，ICLR，KDD，WWW，SIGIR，WSDM，RecSys

25 fan zr Page 3

12/9/2019 4:00

26 fan zr Page 3

12/9/2019 4:01

27 fan zr Page 3

12/9/2019 4:01

28 fan zr Page 3

12/9/2019 4:01

29 fan zr Page 3

12/9/2019 4:01

30 fan zr Page 3

12/9/2019 4:01

31 fan zr Page 3

12/9/2019 4:01

32 fan zr Page 3

12/9/2019 4:01

33 fan zr Page 3

12/9/2019 4:06

本文宗旨：

1. 系统回顾基于深度学习的推荐模型并对当前工作分类
2. 总览当前的最先进模型技术
3. 讨论当前挑战和开放问题，辨明当前趋势和未来发展方向，拓展深度学习推荐系统研究视野

34 fan zr Page 3

12/9/2019 4:02

35 fan zr Page 3

12/9/2019 4:02

36 fan zr Page 3

12/9/2019 4:03

37 fan zr Page 3

12/9/2019 4:09

38 fan zr Page 3

12/9/2019 4:09

第二部分：介绍推荐系统和深度神经网络，并讨论基于深度学习的推荐系统的优缺点，P3
第三部分：提出一个分类框架，并介绍业界最优模型，P7
第四部分：讨论当前的挑战，P25
第五部分：总结论文，P28

39 fan zr Page 3

12/9/2019 4:07

40 fan zr Page 3

12/9/2019 4:07

41 fan zr Page 3

12/9/2019 4:07

42 fan zr Page 3

12/9/2019 4:07

43 fan zr Page 3

12/9/2019 4:12

44 fan zr Page 3

12/9/2019 4:13

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2017). Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Surveys, 1(1), 1–35. Retrieved from http://arxiv.org/abs/1707.07435

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2017). Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Surveys, 1(1), 1–35. Retrieved from http://arxiv.org/abs/1707.07435

45 fan zr Page 3

12/9/2019 6:07

2010. Recommender systems: an introduction.

46 fan zr Page 3

12/9/2019 6:07

47 fan zr Page 4

12/9/2019 6:09

48 fan zr Page 4

12/9/2019 6:51

49 fan zr Page 4

12/9/2019 6:51

50 fan zr Page 4

12/9/2019 6:52

51 fan zr Page 4

12/9/2019 7:08

52 fan zr Page 4

12/9/2019 7:11

53 fan zr Page 4

12/9/2019 7:12

2014. Deep learning: methods and applications.

54 fan zr Page 4

12/9/2019 7:45

55 fan zr Page 4

12/9/2019 7:53

56 fan zr Page 4

12/9/2019 7:48

2012. Marginalized denoising autoencoders for domain adaptation.

2016. Deep Learning. MIT Press.

57 fan zr Page 4

12/9/2019 7:48

58 fan zr Page 4

12/9/2019 7:53

59 fan zr Page 4

12/9/2019 7:49

60 fan zr Page 4

12/9/2019 7:53

61 fan zr Page 4

12/9/2019 7:54

62 fan zr Page 4

12/9/2019 7:54

63 fan zr Page 4

12/9/2019 7:54

64 fan zr Page 4

12/9/2019 7:54

65 fan zr Page 4

12/9/2019 7:54

66 fan zr Page 4

12/9/2019 7:55

2011. The neural autoregressive distribution estimator.

2016. Neural autoregressive distribution
estimation.

67 fan zr Page 4

12/9/2019 7:55

68 fan zr Page 4

12/9/2019 7:56

2014. Generative adversarial nets. In NIPS. 2672–2680.

69 fan zr Page 4

12/9/2019 7:56

70 fan zr Page 4

12/9/2019 7:57

71 fan zr Page 4

12/9/2019 7:57

72 fan zr Page 4

12/9/2019 7:57

73 fan zr Page 4

12/9/2019 7:57

74 fan zr Page 4

12/9/2019 7:58

注意力机制在图像和nlp领域日趋普遍，并且也开始在推荐领域崭露头角

75 fan zr Page 5

12/9/2019 7:59

2015. Human-level control through deep reinforcement learning. Nature

76 fan zr Page 5

12/9/2019 7:59

77 fan zr Page 5

12/9/2019 8:01

78 fan zr Page 5

12/9/2019 8:16

79 fan zr Page 5

12/9/2019 8:18

2016. Deep Learning. MIT Press.

80 fan zr Page 5

13/9/2019 3:39

为什么推荐系统要引入深度神经网络？
1. 深度神经网络结构是端到端的微分，根据输入提供归纳偏向
2. 会话或点击日志的序列结构非常适合循环或卷积网络的归纳偏向
3.多层神经网络由一个个的微分函数组成并可以端到端训练，这一点尤其适合基于内容的推荐
4. 神经网络如CNN，RNN对于多元数据处理非常适合

81 fan zr Page 5

12/9/2019 8:20

82 fan zr Page 5

12/9/2019 8:33

83 fan zr Page 5

12/9/2019 8:33

84 fan zr Page 5

12/9/2019 8:33

85 fan zr Page 5

12/9/2019 8:34

86 fan zr Page 5

12/9/2019 8:35

2015. Session-based recommendations with recurrent
neural networks.

2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In WSDM.

2017. Recurrent recommender networks. In WSDM.

87 fan zr Page 5

13/9/2019 3:32

88 fan zr Page 5

13/9/2019 3:33

89 fan zr Page 5

13/9/2019 3:34

2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In WSDM.

2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In IJCAI.

90 fan zr Page 5

13/9/2019 3:34

91 fan zr Page 5

13/9/2019 3:34

92 fan zr Page 5

13/9/2019 3:35

93 fan zr Page 5

13/9/2019 3:37

94 fan zr Page 5

13/9/2019 3:38

95 fan zr Page 5

13/9/2019 3:38

2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In CIKM.

96 fan zr Page 5

13/9/2019 4:04

针对只有交互的场景，如矩阵补全或协同排序问题，高度复杂又有海量训练实例，深度神经网络非常适合

97 fan zr Page 5

13/9/2019 4:04

2017. Neural collaborative filtering. In WWW.

98 fan zr Page 5

13/9/2019 4:02

99 fan zr Page 5

13/9/2019 4:05

2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In WWW.

尽管神经网络模型表现的更好，在只有交互数据的场景下，当使用基于动量的梯度下降训练时，标准机器学习模型如BPR，MF和CML也表现的相当好了

当然也可以认为这些模型是神经网络结构的，因为它们也利用了深度学习的优势如Adam，Dropout或Batch Normalization

100 fan zr Page 5

13/9/2019 4:05

101 fan zr Page 5

13/9/2019 4:09

2018. NeuRec: On Nonlinear Transformation for
Personalized Ranking.

102 fan zr Page 6

13/9/2019 4:11

2017. Neural Factorization Machines for Sparse Predictive Analytics.

103 fan zr Page 6

13/9/2019 4:10

104 fan zr Page 6

13/9/2019 4:12

深度学习的优势：

1.

非线性转换：矩阵分解，因子分解机，稀疏线性模型，本质都是线性，比较简化，且限制了模型的表达能力，而深度神经网络可以以任意精度近似任意连续函数

2. 表示学习：从输入数据中挖掘隐含的有效因子和有用的表示

3. 序列建模：RNN和CNN都是序列建模的利器，对于用户行为的短时动态变化和物品演化是非常适合的

4. 灵活性

105 fan zr Page 6

13/9/2019 4:17

106 fan zr Page 6

13/9/2019 4:17

107 fan zr Page 6

13/9/2019 4:17

108 fan zr Page 6

13/9/2019 4:17

109 fan zr Page 6

13/9/2019 4:18

2017. Neural Factorization Machines for Sparse Predictive Analytics.

110 fan zr Page 6

13/9/2019 4:17

111 fan zr Page 6

13/9/2019 4:19

112 fan zr Page 6

13/9/2019 4:20

1991. Approximation capabilities of multilayer feedforward networks. Neural networks

1989. Multilayer feedforward networks are universal approximators.

113 fan zr Page 6

13/9/2019 4:20

114 fan zr Page 6

13/9/2019 4:28

115 fan zr Page 6

13/9/2019 6:08

116 fan zr Page 6

13/9/2019 6:08

117 fan zr Page 6

13/9/2019 6:10

使用深度神经网络进行表示学习有两个优势：

1. 减少手工特征工程
2. 推荐模型能够处理多种信息形式，如文本，图像，音频和视频

118 fan zr Page 6

13/9/2019 6:09

119 fan zr Page 6

13/9/2019 6:09

120 fan zr Page 6

13/9/2019 6:12

121 fan zr Page 6

13/9/2019 6:12

122 fan zr Page 6

13/9/2019 6:12

123 fan zr Page 6

13/9/2019 6:12

124 fan zr Page 6

13/9/2019 6:13

125 fan zr Page 6

13/9/2019 6:15

126 fan zr Page 7

13/9/2019 6:15

127 fan zr Page 7

13/9/2019 6:16

推荐系统使用深度学习可能带来的弊端：

1. 可解释性
2. 数据量级要求
3. 超参数调试，不仅深度学习需要，机器学习也需要

128 fan zr Page 7

13/9/2019 6:18

129 fan zr Page 7

13/9/2019 6:19

130 fan zr Page 7

13/9/2019 6:20

2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Recsys.

2018. Multi-Pointer Co-Attention Networks for Recommendation. In SIGKDD.

2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks.

131 fan zr Page 7

13/9/2019 6:22

132 fan zr Page 7

13/9/2019 6:23

133 fan zr Page 7

13/9/2019 6:23

134 fan zr Page 7

13/9/2019 6:23

135 fan zr Page 7

13/9/2019 6:24

2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In WWW.

2017. Collaborative metric learning. WWW

136 fan zr Page 7

13/9/2019 6:26

推荐领域的深度学习模型：业界最优

137 fan zr Page 7

13/9/2019 6:26

138 fan zr Page 7

14/9/2019 1:17

139 fan zr Page 7

14/9/2019 1:19

深度学习推荐模型的分类：

1.

具有神经构建模块的推荐，根据融合的模型不同，分为八类，MLP（适合挖掘用户和物品间的非线性关系），AE，CNNs（提取输入数据的局部和全局表示），RNNs（挖掘时序变化和序列演化），RBM，NADE，AM，AN和DRL

2. 深度混合模型

140 fan zr Page 7

14/9/2019 1:18

141 fan zr Page 7

14/9/2019 1:19

142 fan zr Page 7

14/9/2019 1:20

143 fan zr Page 7

14/9/2019 1:20

144 fan zr Page 7

14/9/2019 1:20

14/9/2019 1:24

基于MLP的推荐模型：

2：

2017. Examples-Rules Guided Deep Neural Network for Makeup Recommendation. In AAAI.

13：

2017. Locally Connected Deep Learning Framework for Industrial-scale Recommender Systems. In WWW.

20：

2016. Wide & deep learning for recommender systems. In Recsys.

27：

2016. Deep neural networks for youtube recommendations. In Recsys.

38：

2017. Neural Citation Network for Context-Aware Citation Recommendation.

47：

2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In IJCAI. 2782–2788.

53：

2017. Neural collaborative filtering. In WWW.

54：

2017. Neural Factorization Machines for Sparse Predictive Analytics.

66：

2015. A Neural Probabilistic Model for Context Based Citation Recommendation. In AAAI.

92：

2017. CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. In WWW.

95：

2015. Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks.. In ISMIR.

157：

2016. Exploring Deep Space: Learning Personalized Ranking in a Semantic Space. In Recsys.

166：

2017. Item Silk Road: Recommending Items from Information Domains to Social Users. (2017).

185：

2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In SIGKDD.

12:

2017. Location-Aware Personalized News Recommendation With Deep Semantic Analysis. IEEE

39:

2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In WWW.

93:

2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems.

112:

2018. Neural Personalized Ranking for Image Recommendation.

134:

14/9/2019 1:53
基于CNN的推荐模型
25：
2017. A hybrid recommendation system considering visual information for predicting favorite restaurants.
WWWJ (2017),
49：
2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In WWW.
50：
2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In AAAI.
75：
2016. Convolutional matrix factorization for document
context-aware recommendation. In Recsys.
76：
2017. Deep Hybrid Recommender Systems via Exploiting Document
Context and Statistics of Items. Information Sciences (2017).
98：
2017. DeepStyle: Learning User Preferences for Visual Recommendation.
105：
2015. Image-based recommendations on styles and
substitutes. In SIGIR.
127：
2017. Representation Learning of Users and Items for Review Rating Prediction
Using Attention-based Convolutional Neural Network. In MLRec.
130：
2016. Automatic Recommendation Technology for Learning
Resources with Convolutional Neural Network. In International Symposium on Educational Technology.
153：
2013. Deep content-based music recommendation. In NIPS.
165：
2017. What Your Images Reveal: Exploiting Visual
Contents for Point-of-Interest Recommendation. In WWW.
172：
2016. Visual background recommendation for dance
performances using dancer-shared images.
202：
2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In
WSDM.
206：
2016. Applying visual user interest profiles for recommendation & personalisation.

6：
2017. Graph convolutional matrix completion.
44：
2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In IJCAI.
51：
2018. Outer Product-based Neural Collaborative
Filtering.
83：
2018. Collaborative Deep Metric Learning for
Video Understanding. (2018).

147 fan zr Page 8

14/9/2019 1:20

148 fan zr Page 8

14/9/2019 1:20

149 fan zr Page 8

14/9/2019 1:20

150 fan zr Page 8

14/9/2019 2:21

151 fan zr Page 8

14/9/2019 2:22

152 fan zr Page 8

14/9/2019 2:22

153 fan zr Page 8

14/9/2019 2:23

131：

2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. ACMComputing

154 fan zr Page 9

14/9/2019 2:44

多层感知机模型是大量高级模型的基础并被广泛应用于很多领域

155 fan zr Page 9

14/9/2019 16:35

156 fan zr Page 9

14/9/2019 16:35

157 fan zr Page 9

14/9/2019 16:39
37：
2015. Neural network matrix factorization.
53：
2017. Neural collaborative filtering. In WWW.

158 fan zr Page 9

14/9/2019 16:38

159 fan zr Page 9

14/9/2019 16:39

160 fan zr Page 9

14/9/2019 16:39

161 fan zr Page 9

14/9/2019 17:11

162 fan zr Page 9

14/9/2019 17:11

163 fan zr Page 9

14/9/2019 17:11

164 fan zr Page 10

14/9/2019 17:12

165 fan zr Page 10

14/9/2019 17:14
如何进行负采样，需调研一下

166 fan zr Page 10

14/9/2019 17:13

167 fan zr Page 10

15/9/2019 14:28

92:
2017. CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. In WWW.
166:
2017. Item Silk Road: Recommending Items from Information Domains to Social Users. (2017).

168 fan zr Page 10

15/9/2019 14:28

112:
2018. Neural Personalized Ranking for Image Recommendation.
134:
2018. Neural Collaborative Ranking.

169 fan zr Page 10

15/9/2019 14:29

184:
2017. Deep Matrix Factorization Models for Recommender Systems.. In IJCAI.
195:
2018. NeuRec: On Nonlinear Transformation for Personalized Ranking.

170 fan zr Page 10

15/9/2019 14:29

171 fan zr Page 10

15/9/2019 14:30

172 fan zr Page 10

15/9/2019 14:31

173 fan zr Page 10

15/9/2019 14:31

47:
2017. DeepFM: A Factorization-Machine based Neural Network
119:
2010. Factorization Machines. In 2010 IEEE International Conference

174 fan zr Page 10

15/9/2019 14:53

175 fan zr Page 10

15/9/2019 14:57

176 fan zr Page 10

15/9/2019 14:58

177 fan zr Page 10

15/9/2019 14:58
93:
2018. xDeepFM: Combining Explicit
and Implicit Feature Interactions for Recommender Systems.
54:
2017. Neural Factorization Machines for Sparse Predictive Analytics.

178 fan zr Page 10

15/9/2019 14:59

179 fan zr Page 10

15/9/2019 14:59

180 fan zr Page 11

15/9/2019 15:00
兼具记忆性和泛化性

181 fan zr Page 11

15/9/2019 15:00

182 fan zr Page 11

15/9/2019 15:00

183 fan zr Page 11

15/9/2019 15:02

184 fan zr Page 11

15/9/2019 15:02

185 fan zr Page 11

15/9/2019 15:02

186 fan zr Page 11

15/9/2019 15:03

187 fan zr Page 11

15/9/2019 15:04

188 fan zr Page 11

15/9/2019 15:04

189 fan zr Page 11

15/9/2019 23:40
13:
2017. Locally Connected Deep Learning Framework for
Industrial-scale Recommender Systems. In WWW.

190 fan zr Page 11

15/9/2019 23:40

191 fan zr Page 11

15/9/2019 23:41

192 fan zr Page 11

15/9/2019 23:41
27:
2016. Deep neural networks for youtube recommendations. In Recsys.

193 fan zr Page 11

15/9/2019 23:43

194 fan zr Page 11

15/9/2019 23:44

2:

2017. Examples-Rules Guided Deep Neural Network for Makeup Recommendation. In AAAI.

195 fan zr Page 11

17/9/2019 1:44

196 fan zr Page 11

17/9/2019 1:44

197 fan zr Page 11

17/9/2019 1:44

60:

2017. Collaborative metric learning.

198 fan zr Page 11

17/9/2019 1:45

199 fan zr Page 11

17/9/2019 1:45

200 fan zr Page 11

17/9/2019 1:46

65:

2013. Learning deep structured semantic models for web search using clickthrough data. In CIKM.

39:

2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In WWW.

182:

2016. Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. In CIKM.

201 fan zr Page 11

17/9/2019 1:46

202 fan zr Page 11

17/9/2019 3:23

203 fan zr Page 12

17/9/2019 3:24

182:

2016. Tag-aware personalized recommendation
using a deep-semantic similarity model with negative sampling. In CIKM.

204 fan zr Page 12

17/9/2019 3:24

205 fan zr Page 12

17/9/2019 3:25

206 fan zr Page 12

17/9/2019 3:25

207 fan zr Page 12

17/9/2019 3:26

208 fan zr Page 12

17/9/2019 3:26

209 fan zr Page 12

17/9/2019 3:27

183:

2017. Tag-aware personalized recommendation
using a hybrid deep model. (2017).

210 fan zr Page 12

17/9/2019 3:28

211 fan zr Page 12

17/9/2019 3:28

212 fan zr Page 12

17/9/2019 3:29

213 fan zr Page 12

17/9/2019 3:30

214 fan zr Page 12

17/9/2019 3:30

215 fan zr Page 12

17/9/2019 3:30

216 fan zr Page 12

17/9/2019 3:31

217 fan zr Page 12

17/9/2019 3:31

218 fan zr Page 12

17/9/2019 3:31

125:
2015. Autorec: Autoencoders meet collaborative filtering. In WWW.

219 fan zr Page 12

17/9/2019 3:33

220 fan zr Page 13

17/9/2019 3:33

221 fan zr Page 13

17/9/2019 3:43

222 fan zr Page 13

17/9/2019 3:46

223 fan zr Page 13

17/9/2019 3:46

224 fan zr Page 13

17/9/2019 3:46

225 fan zr Page 13

17/9/2019 3:47

226 fan zr Page 13

17/9/2019 3:47

227 fan zr Page 13

17/9/2019 6:11

136:

2016. Hybrid Recommender System based on Autoencoders.

137:

2015. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In NIPS

228 fan zr Page 13

17/9/2019 6:12

229 fan zr Page 13

17/9/2019 6:12

230 fan zr Page 13

17/9/2019 6:13

231 fan zr Page 13

17/9/2019 6:13

232 fan zr Page 13

17/9/2019 6:13

233 fan zr Page 13

17/9/2019 6:14

234 fan zr Page 14

17/9/2019 6:14

235 fan zr Page 14

17/9/2019 6:15

236 fan zr Page 14

17/9/2019 6:14

177:

2016. Collaborative denoising auto-encoders for top-n recommender systems. In WSDM.

237 fan zr Page 14

17/9/2019 6:15

238 fan zr Page 14

17/9/2019 6:16

239 fan zr Page 14

17/9/2019 6:16

240 fan zr Page 14

17/9/2019 6:26

241 fan zr Page 14

17/9/2019 6:26

242 fan zr Page 14

17/9/2019 6:26

243 fan zr Page 14

17/9/2019 6:27

244 fan zr Page 14

17/9/2019 6:27

245 fan zr Page 14

17/9/2019 6:27

246 fan zr Page 14

17/9/2019 6:27

247 fan zr Page 14

17/9/2019 6:28

94 :

2018. Variational Autoencoders for Collaborative Filtering.

248 fan zr Page 14

17/9/2019 7:17

249 fan zr Page 14

17/9/2019 7:18

250 fan zr Page 14

17/9/2019 7:18

251 fan zr Page 14

17/9/2019 7:18

252 fan zr Page 14

17/9/2019 7:18

114:

2014. Autoencoder-based collaborative filtering.

253 fan zr Page 14

17/9/2019 8:17

254 fan zr Page 14

17/9/2019 8:18

255 fan zr Page 14

17/9/2019 8:18

256 fan zr Page 14

17/9/2019 8:18

257 fan zr Page 14

17/9/2019 8:18

258 fan zr Page 14

17/9/2019 8:19

259 fan zr Page 14

17/9/2019 8:19

260 fan zr Page 14

17/9/2019 8:20

261 fan zr Page 14

17/9/2019 8:19

159：

2015. Collaborative deep learning for recommender systems. In SIGKDD.

262 fan zr Page 14

17/9/2019 8:21

161：

2016. Towards Bayesian deep learning: A framework and some existing methods.

263 fan zr Page 14

17/9/2019 8:20

264 fan zr Page 14

17/9/2019 8:21

265 fan zr Page 14

17/9/2019 8:21

266 fan zr Page 14

17/9/2019 8:21

267 fan zr Page 14

17/9/2019 8:22

268 fan zr Page 15

17/9/2019 9:08

269 fan zr Page 15

17/9/2019 9:09

270 fan zr Page 15

17/9/2019 9:09

63:

2008. Collaborative Filtering for Implicit Feedback Datasets.

161:

2016. Towards Bayesian deep learning: A framework and some existing methods.

271 fan zr Page 15

17/9/2019 9:10

272 fan zr Page 15

17/9/2019 9:10

158:

2015. Relational Stacked Denoising Autoencoder for Tag Recommendation..

273 fan zr Page 15

17/9/2019 9:10

274 fan zr Page 15

17/9/2019 9:10

275 fan zr Page 15

17/9/2019 9:11

89:

2017. Collaborative Variational Autoencoder for Recommender Systems. In SIGKDD.

276 fan zr Page 15

17/9/2019 9:12

277 fan zr Page 15

17/9/2019 9:12

278 fan zr Page 15

17/9/2019 9:12

279 fan zr Page 15

17/9/2019 9:12

188:
2016. Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback. In PAKDD.

280 fan zr Page 15

17/9/2019 9:26

281 fan zr Page 15

17/9/2019 9:26

120:
2009. BPR: Bayesian personalized ranking from implicit feedback.
177:
2016. Collaborative denoising auto-encoders for top-n recommender systems. In WSDM.
188:
2016. Collaborative deep ranking: a hybrid pair-wise recommendation algorithm with implicit feedback. In PAKDD.

282 fan zr Page 15

17/9/2019 9:27

283 fan zr Page 15

17/9/2019 9:28

284 fan zr Page 15

17/9/2019 9:29

285 fan zr Page 15

17/9/2019 9:29

286 fan zr Page 15

17/9/2019 9:29

88:

2015. Deep collaborative filtering via marginalized denoising auto-encoder.

287 fan zr Page 15

17/9/2019 9:30

153:

2013. Deep content-based music recommendation. In NIPS.

159:

2015. Collaborative deep learning for recommender systems. In SIGKDD.

167:

2014. Improving content-based and hybrid music recommendation using deep learning.

288 fan zr Page 15

17/9/2019 9:32

289 fan zr Page 15

17/9/2019 9:32

290 fan zr Page 15

17/9/2019 9:32

291 fan zr Page 15

17/9/2019 9:32

292 fan zr Page 15

17/9/2019 9:40

15:

2012. Marginalized denoising autoencoders for domain adaptation.

293 fan zr Page 16

17/9/2019 9:41

294 fan zr Page 16

17/9/2019 9:41

295 fan zr Page 16

17/9/2019 11:53

296 fan zr Page 16

17/9/2019 11:56

297 fan zr Page 16

17/9/2019 11:53

196:
2017. AutoSVD++: An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders.
122:
2011. Contractive auto-encoders: Explicit invariance
during feature extraction. In ICML.
79:
2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In SIGKDD.

298 fan zr Page 16

17/9/2019 11:56

299 fan zr Page 16

17/9/2019 11:56

300 fan zr Page 16

17/9/2019 11:56

301 fan zr Page 16

17/9/2019 11:56

302 fan zr Page 16

17/9/2019 11:56

303 fan zr Page 16

17/9/2019 11:56
170:
2016. Collaborative filtering and deep learning based hybrid recommendation for cold start problem. IEEE,
171:
2017. Collaborative filtering and deep learning based recommendation system for cold start items.
80:
2010. Collaborative filtering with temporal dynamics.

304 fan zr Page 16

17/9/2019 11:58

305 fan zr Page 16

17/9/2019 11:58

306 fan zr Page 16

17/9/2019 11:59

307 fan zr Page 16

17/9/2019 11:59

308 fan zr Page 16

17/9/2019 11:59

309 fan zr Page 16

17/9/2019 11:59

310 fan zr Page 16

17/9/2019 13:38

311 fan zr Page 16

17/9/2019 13:38

312 fan zr Page 16

17/9/2019 12:02

165:
2017. What Your Images Reveal: Exploiting Visual
Contents for Point-of-Interest Recommendation. In WWW.
25:
2017. A hybrid recommendation system considering visual information for predicting favorite restaurants.
WWWJ (2017),
50:
2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In AAAI.
49:
2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In WWW.
191:
2018. Aesthetic-based clothing recommendation. In
WWW.
110:
2017. Personalized Deep Learning
for Tag Recommendation.
84:
2016. Comparative Deep Learning of Hybrid Representations for
Image Recommendations.
118:
2016. ConTagNet: exploiting user context for image tag recommendation. In Proceedings
ofthe 2016 ACMon

313 fan zr Page 16

17/9/2019 13:39

314 fan zr Page 16

17/9/2019 13:39

315 fan zr Page 16

17/9/2019 13:40

316 fan zr Page 16

17/9/2019 13:40

317 fan zr Page 16

17/9/2019 13:41

318 fan zr Page 16

17/9/2019 13:41

319 fan zr Page 16

17/9/2019 13:47

320 fan zr Page 16

17/9/2019 13:47

321 fan zr Page 16

17/9/2019 13:48

322 fan zr Page 16

17/9/2019 13:48

323 fan zr Page 16

17/9/2019 13:49

324 fan zr Page 16

17/9/2019 13:49

325 fan zr Page 16

17/9/2019 12:00

326 fan zr Page 16

17/9/2019 13:51

327 fan zr Page 16

17/9/2019 13:52

328 fan zr Page 16

17/9/2019 13:52

329 fan zr Page 16

17/9/2019 13:52

330 fan zr Page 17

17/9/2019 13:51

202:
2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In WSDM.
11:
2017. Transnets: Learning to transform for recommendation. In Recsys.
130:
2016. Automatic Recommendation Technology for Learning Resources with Convolutional Neural Network. In International Symposium on Educational Technology.
153:
2013. Deep content-based music recommendation. In NIPS.
75:
2016. Convolutional matrix factorization for document context-aware recommendation. In Recsys.
148:
2017. 3D Convolutional Networks for Session-based Recommendation with Content Features. In Recsys.

331 fan zr Page 17

17/9/2019 13:53

332 fan zr Page 17

17/9/2019 13:53

333 fan zr Page 17

17/9/2019 13:56

334 fan zr Page 17

17/9/2019 13:57

335 fan zr Page 17

17/9/2019 14:01

336 fan zr Page 17

17/9/2019 14:02

337 fan zr Page 17

17/9/2019 14:02

338 fan zr Page 17

17/9/2019 12:00

339 fan zr Page 17

17/9/2019 14:03

153:

2013. Deep content-based music recommendation. In NIPS.

83:

2018. Collaborative Deep Metric Learning for
Video Understanding. (2018).

340 fan zr Page 17

17/9/2019 14:15

341 fan zr Page 17

17/9/2019 14:16

342 fan zr Page 17

17/9/2019 14:16

343 fan zr Page 17

18/9/2019 1:23

344 fan zr Page 17

18/9/2019 1:24

51:

2018. Outer Product-based Neural Collaborative
Filtering.

143:

2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In WSDM.

345 fan zr Page 17

18/9/2019 1:27

346 fan zr Page 17

18/9/2019 1:27

347 fan zr Page 17

18/9/2019 1:27

348 fan zr Page 17

18/9/2019 1:27

349 fan zr Page 17

18/9/2019 1:28

350 fan zr Page 17

18/9/2019 1:28

351 fan zr Page 17

18/9/2019 1:29

352 fan zr Page 17

18/9/2019 1:29

353 fan zr Page 17

18/9/2019 1:29

354 fan zr Page 17

18/9/2019 1:29
77:
2016. Semi-supervised classification with graph convolutional networks.
6:
2017. Graph convolutional matrix completion.
190:
2018. Graph Convolutional Neural
Networks for Web-Scale Recommender Systems.

355 fan zr Page 17

18/9/2019 1:36

356 fan zr Page 17

18/9/2019 1:37

357 fan zr Page 17

18/9/2019 1:39

358 fan zr Page 17

18/9/2019 1:40

359 fan zr Page 17

18/9/2019 1:40

360 fan zr Page 18

18/9/2019 1:41

361 fan zr Page 18

18/9/2019 1:41

362 fan zr Page 18

18/9/2019 1:41

56:
2015. Session-based recommendations with recurrent
neural networks.
142:
2016. Improved recurrent neural networks for session-based recommendations. In Recsys.
176:
2016. Personal recommendation using deep
recurrent neural networks in NetEase. In ICDE.

363 fan zr Page 18

18/9/2019 1:42

364 fan zr Page 18

18/9/2019 1:49

365 fan zr Page 18

18/9/2019 1:50

366 fan zr Page 18

18/9/2019 1:50

367 fan zr Page 18

18/9/2019 1:51

368 fan zr Page 18

18/9/2019 1:52

369 fan zr Page 18

18/9/2019 1:53

370 fan zr Page 18

18/9/2019 1:53

371 fan zr Page 18

18/9/2019 1:53

372 fan zr Page 18

18/9/2019 1:53

373 fan zr Page 18

18/9/2019 1:54

55:
2017. Recurrent neural networks with top-k gains for session-based recommendations.
56:
2015. Session-based recommendations with recurrent
neural networks.

374 fan zr Page 18

18/9/2019 1:54

375 fan zr Page 18

18/9/2019 1:55

376 fan zr Page 18

18/9/2019 1:55

377 fan zr Page 18

18/9/2019 1:59

378 fan zr Page 18

18/9/2019 1:59

379 fan zr Page 18

18/9/2019 6:28

380 fan zr Page 18

18/9/2019 6:28

381 fan zr Page 18

18/9/2019 1:59

176:
2016. Personal recommendation using deep
recurrent neural networks in NetEase. In ICDE.
117:
2017. Personalizing session-based recommendations
with hierarchical recurrent neural networks. In Recsys.

382 fan zr Page 18

18/9/2019 6:29

383 fan zr Page 18

18/9/2019 6:29

384 fan zr Page 18

18/9/2019 6:30

57：
2016. Parallel recurrent neural network architectures
for feature-rich session-based recommendations. In Recsys.
132：
2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks.

385 fan zr Page 18

18/9/2019 6:33

386 fan zr Page 18

18/9/2019 6:36

387 fan zr Page 18

18/9/2019 6:36

388 fan zr Page 18

18/9/2019 6:36

389 fan zr Page 18

18/9/2019 6:30

390 fan zr Page 18

18/9/2019 6:36

391 fan zr Page 18

18/9/2019 6:38

392 fan zr Page 18

18/9/2019 6:38

393 fan zr Page 18

18/9/2019 6:38

394 fan zr Page 18

18/9/2019 6:38

395 fan zr Page 18

18/9/2019 6:38

68：
2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommenda-tion. In Recsys.
103：
2018. Evaluation of Session-based Recommendation Algorithms.

396 fan zr Page 18

18/9/2019 6:39

397 fan zr Page 18

18/9/2019 6:39

398 fan zr Page 18

18/9/2019 6:44

基于会话的推荐不会感知用户标识，而序列化推荐是需要用户身份的

399 fan zr Page 18

18/9/2019 6:40

400 fan zr Page 18

18/9/2019 6:43

401 fan zr Page 18

18/9/2019 6:43

402 fan zr Page 18

18/9/2019 6:43

175：

2017. Recurrent recommender networks. In WSDM.

403 fan zr Page 18

18/9/2019 7:20

404 fan zr Page 18

18/9/2019 7:20

405 fan zr Page 19

18/9/2019 15:23

406 fan zr Page 19

18/9/2019 15:23

407 fan zr Page 19

18/9/2019 15:24

408 fan zr Page 19

18/9/2019 15:25

174：

2016. Joint Training of Ratings and Reviews with Recurrent
Recommender Networks.

127：

2017. Representation Learning of Users and Items for Review Rating Prediction
Using Attention-based Convolutional Neural Network. In MLRec.

202：

2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In WSDM.

87：

2017. Neural Rating Regression with Abstractive Tips Generation for
Recommendation.

73：

2017. Neural survival recommender. In WSDM.

91：

2018. Learning from History and Present: Next-item
Recommendation via Discriminatively Exploiting User Behaviors. In SIGKDD.

24：

2018. Q&R: A Two-Stage Approach toward Interactive
Recommendation. In SIGKDD.

35：

2017. Sequential user-based recurrent neural network recommendations. In Recsys.

409 fan zr Page 19

18/9/2019 15:26

410 fan zr Page 19

18/9/2019 15:30

411 fan zr Page 19

18/9/2019 15:39

412 fan zr Page 19

18/9/2019 15:40

413 fan zr Page 19

18/9/2019 15:40

414 fan zr Page 19

18/9/2019 15:40

415 fan zr Page 19

18/9/2019 15:41

416 fan zr Page 19

18/9/2019 15:41

417 fan zr Page 19

18/9/2019 15:41

418 fan zr Page 19

18/9/2019 15:42

419 fan zr Page 19

18/9/2019 15:42

420 fan zr Page 19

18/9/2019 15:42

421 fan zr Page 19

18/9/2019 15:43

422 fan zr Page 19

18/9/2019 15:43

29：

2016. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In Recsys.

423 fan zr Page 19

18/9/2019 15:44

424 fan zr Page 20

18/9/2019 15:44

425 fan zr Page 20

18/9/2019 15:57

426 fan zr Page 20

18/9/2019 15:45

5 :

2016. Ask the gru: Multi-task learning for deep text recommendations. In Proceedings ofthe 10th ACM Conference on Recommender Systems.

427 fan zr Page 20

18/9/2019 15:57

428 fan zr Page 20

18/9/2019 15:57

429 fan zr Page 20

18/9/2019 15:57

430 fan zr Page 20

18/9/2019 15:58

431 fan zr Page 20

18/9/2019 15:58

113 :

2017. Embedding-based News Recommendation for Millions of Users. In SIGKDD.

432 fan zr Page 20

18/9/2019 16:00

433 fan zr Page 20

18/9/2019 16:00

87：

2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation.

434 fan zr Page 20

18/9/2019 16:02

435 fan zr Page 20

18/9/2019 16:02

436 fan zr Page 20

18/9/2019 16:02

437 fan zr Page 20

18/9/2019 16:02

438 fan zr Page 20

18/9/2019 16:03

439 fan zr Page 20

18/9/2019 16:03

135：

2016. Multi-rate deep learning for temporal recommendation. In SIGIR.

440 fan zr Page 20

18/9/2019 16:04

441 fan zr Page 20

19/9/2019 1:17

442 fan zr Page 20

18/9/2019 16:04

123：

2007. Restricted Boltzmann machines for collaborative filtering. In ICML.

443 fan zr Page 20

19/9/2019 1:17

444 fan zr Page 20

19/9/2019 1:19

445 fan zr Page 20

19/9/2019 1:46

446 fan zr Page 20

19/9/2019 1:46

447 fan zr Page 20

19/9/2019 1:47

45:

2016. Deep Learning. MIT Press.

448 fan zr Page 20

19/9/2019 1:47

449 fan zr Page 20

19/9/2019 1:47

450 fan zr Page 20

19/9/2019 1:48

451 fan zr Page 20

19/9/2019 1:49

42:

2013. A non-iid framework for collaborative filtering with restricted boltzmann machines. In ICML.

100:

2015. Item Category Aware Conditional Restricted Boltzmann Machine Based Recommendation.

452 fan zr Page 20

19/9/2019 1:57

453 fan zr Page 20

19/9/2019 1:58

454 fan zr Page 21

19/9/2019 1:58

455 fan zr Page 21

19/9/2019 1:59

456 fan zr Page 21

19/9/2019 3:46

457 fan zr Page 21

19/9/2019 3:46

458 fan zr Page 21

19/9/2019 3:47
3：
2014. Multiple object recognition with visual attention.
104：
2015. Effective approaches to attention-based neural machine translation.
155：
2017.Attention is all you need. In Advances in Neural Information Processing Systems.
22：
2014. End-to-end continuous speech recognition using
attention-based recurrent NN: first results.
23：
2015. Attention-based models for speech
recognition. In Advances in Neural Information Processing Systems.
127：
2017. Representation Learning of Users and Items for Review Rating Prediction
Using Attention-based Convolutional Neural Network. In MLRec.
14：
2017. Attentive Collaborative Filtering:
Multimedia Recommendation with Item- and Component-Level Attention.

459 fan zr Page 21

19/9/2019 9:24

460 fan zr Page 21

19/9/2019 9:29

461 fan zr Page 21

19/9/2019 9:29

462 fan zr Page 21

19/9/2019 9:29

463 fan zr Page 21

19/9/2019 9:30

464 fan zr Page 21

19/9/2019 9:30

465 fan zr Page 21

19/9/2019 9:30

466 fan zr Page 21

19/9/2019 9:57

44 :

2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In IJCAI.

467 fan zr Page 21

19/9/2019 9:57

468 fan zr Page 21

19/9/2019 9:58

469 fan zr Page 21

19/9/2019 11:02

470 fan zr Page 21

19/9/2019 11:02

471 fan zr Page 21

19/9/2019 11:02

472 fan zr Page 21

19/9/2019 11:03

473 fan zr Page 21

19/9/2019 11:03

145:

2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In WWW.

70:

2018. Attentive Contextual Denoising Autoencoder for Recommendation.

99:

2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In SIGKDD.

189:

2018. Sequential
Recommender System based on Hierarchical Attention Networks. In IJCAI.

474 fan zr Page 21

19/9/2019 11:05

475 fan zr Page 21

19/9/2019 11:07

476 fan zr Page 21

19/9/2019 11:07

477 fan zr Page 21

19/9/2019 11:07

478 fan zr Page 21

19/9/2019 11:08

90 :

2016. Hashtag recommendation with topical attention-based LSTM. In COLING.

101 :

2017. Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture. In Recsys.

479 fan zr Page 21

19/9/2019 11:08

480 fan zr Page 21

19/9/2019 11:09

481 fan zr Page 22

19/9/2019 11:09

482 fan zr Page 22

19/9/2019 11:13

483 fan zr Page 22

19/9/2019 11:17

44：
2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In IJCAI.
127：
2017. Representation Learning of Users and Items for Review Rating Prediction
Using Attention-based Convolutional Neural Network. In MLRec.
169：
2017. Dynamic Attention Deep Model for
Article Recommendation by Learning Human Editorsfi Demonstration. In SIGKDD.

484 fan zr Page 22

19/9/2019 14:18

485 fan zr Page 22

19/9/2019 14:18

486 fan zr Page 22

19/9/2019 14:18

487 fan zr Page 22

19/9/2019 14:18

488 fan zr Page 22

19/9/2019 14:18

489 fan zr Page 22

19/9/2019 14:36

490 fan zr Page 22

19/9/2019 14:37

491 fan zr Page 22

19/9/2019 14:37

492 fan zr Page 22

19/9/2019 14:39

194：
2018. Next Item Recommendation with Self-Attention.
205：
2017. ATRank: An Attention-Based
User Behavior Modeling Framework for Recommendation.
146：
2018. Multi-Pointer Co-Attention Networks for Recommendation. In SIGKDD.
193：
Hashtag Recommendation for Multimodal Microblog
Using Co-Attention Network. In IJCAI.
62：
2018. Leveraging Meta-path based Context for Top-N Recommendation with
A Neural Co-Attention Model. In SIGKDD.

493 fan zr Page 22

19/9/2019 14:38

494 fan zr Page 22

19/9/2019 14:43

495 fan zr Page 22

19/9/2019 14:43

496 fan zr Page 22

19/9/2019 14:43

497 fan zr Page 22

19/9/2019 14:44

498 fan zr Page 22

19/9/2019 14:48

499 fan zr Page 22

19/9/2019 14:48

500 fan zr Page 22

19/9/2019 14:48

501 fan zr Page 22

19/9/2019 14:49

502 fan zr Page 22

19/9/2019 14:49

503 fan zr Page 22

19/9/2019 15:03

504 fan zr Page 22

19/9/2019 14:52

81：

2011. The neural autoregressive distribution estimator.

204：

2016. A Neural Autoregressive Approach to Collaborative Filtering. In ICML.

505 fan zr Page 22

19/9/2019 15:04

506 fan zr Page 22

19/9/2019 15:04

507 fan zr Page 22

19/9/2019 15:04

508 fan zr Page 22

19/9/2019 15:10

509 fan zr Page 22

19/9/2019 15:10

510 fan zr Page 22

19/9/2019 15:10

511 fan zr Page 22

19/9/2019 15:10

512 fan zr Page 22

19/9/2019 15:12

513 fan zr Page 22

19/9/2019 15:12

514 fan zr Page 22

19/9/2019 15:14

515 fan zr Page 22

19/9/2019 15:15

516 fan zr Page 22

19/9/2019 15:13

203：
2016. Neural Autoregressive Collaborative Filtering for Implicit Feedback. In Recsys.
36：
2016. Collaborative Filtering with User-Item Co-Autoregressive Models.

517 fan zr Page 23

19/9/2019 15:15

518 fan zr Page 23

19/9/2019 15:17

519 fan zr Page 23

19/9/2019 15:17

520 fan zr Page 23

19/9/2019 15:52

21：
2018. Reinforcement Learning based
Recommender System using Biclustering Technique.
107：
2018. Deep Reinforcement Learning for Recommender
Systems.
168：
2014. Exploration in interactive personalized music recommendation: a reinforcement
learning approach.
198：
2018. Deep Reinforcement Learning for Page-wise
Recommendations.
199：
2018. Recommendations with Negative Feedback
via Pairwise Deep Reinforcement Learning.
200：
2018. DRN: A Deep
Reinforcement Learning Framework for News Recommendation. In WWW.

521 fan zr Page 23

19/9/2019 15:54

522 fan zr Page 23

19/9/2019 15:55

523 fan zr Page 23

19/9/2019 15:56

524 fan zr Page 23

19/9/2019 15:56

525 fan zr Page 23

19/9/2019 15:56

526 fan zr Page 23

19/9/2019 15:56

527 fan zr Page 23

19/9/2019 15:56

528 fan zr Page 23

19/9/2019 15:57

529 fan zr Page 23

19/9/2019 15:57

530 fan zr Page 23

19/9/2019 15:57

531 fan zr Page 23

20/9/2019 1:19

532 fan zr Page 23

20/9/2019 1:18

86:

2010. A contextual-bandit approach to personalized news article recommen-dation.

533 fan zr Page 23

20/9/2019 1:45

534 fan zr Page 23

20/9/2019 1:58

535 fan zr Page 23

20/9/2019 1:46

536 fan zr Page 23

20/9/2019 1:47
162:
2017. IRGAN: A Minimax
Game for Unifying Generative and Discriminative Information Retrieval Models.

537 fan zr Page 23

20/9/2019 1:58

538 fan zr Page 23

20/9/2019 1:59

539 fan zr Page 23

20/9/2019 1:59

540 fan zr Page 23

20/9/2019 2:00

541 fan zr Page 23

20/9/2019 2:00

542 fan zr Page 23

20/9/2019 2:03

543 fan zr Page 23

20/9/2019 2:03

544 fan zr Page 23

20/9/2019 2:06

545 fan zr Page 23

20/9/2019 2:07

546 fan zr Page 23

20/9/2019 2:09

547 fan zr Page 23

20/9/2019 2:10

548 fan zr Page 24

20/9/2019 2:10

549 fan zr Page 24

20/9/2019 3:03

550 fan zr Page 24

20/9/2019 3:04

551 fan zr Page 24

20/9/2019 3:13

552 fan zr Page 24

20/9/2019 3:04

52：
2018. Adversarial Personalized Ranking for Recommendation. In SIGIR.
9：
2018. Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation. In AAAI.
164：
2018. Neural Memory Streaming Recommender
Networks with Adversarial Training. In SIGKDD.

553 fan zr Page 24

20/9/2019 3:14

554 fan zr Page 24

20/9/2019 3:15

555 fan zr Page 24

20/9/2019 3:15

556 fan zr Page 24

20/9/2019 3:16

557 fan zr Page 24

20/9/2019 3:16

558 fan zr Page 24

20/9/2019 3:17

559 fan zr Page 24

20/9/2019 3:17

560 fan zr Page 24

20/9/2019 3:28

561 fan zr Page 24

20/9/2019 3:18
192:
2016. Collaborative knowledge base embedding for
recommender systems. In SIGKDD.
96:
2015. Learning Entity and Relation Embeddings for Knowledge Graph
Completion. In AAAI.

562 fan zr Page 24

20/9/2019 3:32

563 fan zr Page 24

20/9/2019 3:32

564 fan zr Page 24

20/9/2019 6:50

565 fan zr Page 24

20/9/2019 6:52

566 fan zr Page 24

20/9/2019 6:52

567 fan zr Page 24

20/9/2019 3:17

568 fan zr Page 24

20/9/2019 6:55

569 fan zr Page 24

20/9/2019 6:53

82：
2016. Quote Recommendation in Dialogue using Deep Neural
Network. In SIGIR.
193：
Hashtag Recommendation for Multimodal Microblog
Using Co-Attention Network. In IJCAI.
38：
2017. Neural Citation Network for Context-Aware Citation Recommendation.
17：
2017. Personalized Key Frame Recommendation. In
SIGIR.

570 fan zr Page 24

20/9/2019 6:55

571 fan zr Page 24

20/9/2019 6:56

572 fan zr Page 24

20/9/2019 6:59

573 fan zr Page 24

20/9/2019 6:59

574 fan zr Page 24

20/9/2019 6:59

575 fan zr Page 24

20/9/2019 6:59

576 fan zr Page 24

20/9/2019 7:00

577 fan zr Page 24

20/9/2019 7:02

160：

2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In NIPS.

578 fan zr Page 24

20/9/2019 3:18

579 fan zr Page 24

20/9/2019 7:01

580 fan zr Page 24

20/9/2019 7:01

581 fan zr Page 24

20/9/2019 7:01

582 fan zr Page 25

20/9/2019 7:01

583 fan zr Page 25

20/9/2019 3:18

584 fan zr Page 25

20/9/2019 7:02

163：

2018. Supervised Reinforcement Learning with Recurrent Neural Network for Dynamic Treatment Recommendation. In SIGKDD.

585 fan zr Page 25

20/9/2019 7:22

586 fan zr Page 25

20/9/2019 7:22

587 fan zr Page 25

20/9/2019 7:23

588 fan zr Page 25

20/9/2019 7:23

589 fan zr Page 25

20/9/2019 7:23

590 fan zr Page 25

20/9/2019 7:23

591 fan zr Page 25

20/9/2019 7:25

592 fan zr Page 25

20/9/2019 7:25

593 fan zr Page 25

20/9/2019 7:25

20/9/2019 7:25
1:
2005. Toward the next generation of recommender systems: A survey of the
state-of-the-art and possible extensions. IEEE transactions
85:
2015. New Directions in Recommender Systems. In WSDM.
151:
2016. Towards latent context-aware recommendation systems. Knowledge-
92:
2017. CCCFNet: A Content-Boosted Collaborative Filtering Neural
Network for Cross Domain Recommender Systems. In WWW.
196:
2017. AutoSVD++: An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders.
50:
188:
2016. Collaborative deep ranking: a hybrid pair-wise recommendation
algorithm with implicit feedback. In PAKDD.
203:
2016. Neural Autoregressive Collaborative Filtering for Implicit
Feedback. In Recsys.
38:
2017. Neural Citation Network for Context-Aware Citation Recommendation.
75:
2016. Convolutional matrix factorization for document
context-aware recommendation. In Recsys.
118:
2016. ConTagNet: exploiting user context for image tag recommendation. In Proceedings
ofthe 2016 ACMon
149:
2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural
Networks. In
Recsys.
151:
2016. Towards latent context-aware recommendation systems. Knowledge-
87:
2017. Neural Rating Regression with Abstractive Tips Generation for
Recommendation.
127:
2017. Representation Learning of Users and Items for Review Rating Prediction
Using Attention-based Convolutional Neural Network. In MLRec.
174:
2016. Joint Training of Ratings and Reviews with Recurrent
Recommender Networks.
202:
2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In
WSDM.
61:
2016. Immersive recommendation: News and event
recommendations using personal digital traces. In WWW.
186:
2016. Things of interest recommendation by leveraging heterogeneous relations
in the internet of things. ACMTransactions

595 fan zr Page 25

20/9/2019 7:26

596 fan zr Page 25

20/9/2019 7:26

597 fan zr Page 25

20/9/2019 7:27

598 fan zr Page 25

20/9/2019 7:27

599 fan zr Page 25

20/9/2019 7:49

尽管现存模型研究了深度模型在挖掘推荐的用户和物品信息，隐式反馈，上下文信息，评论文本方面的效率，但没有充分利用这些侧面信息，也没有充分挖掘可用数据。
很少有工作研究用户的社交足迹和物理世界

600 fan zr Page 25

20/9/2019 7:27

601 fan zr Page 25

20/9/2019 7:27

602 fan zr Page 25

20/9/2019 7:40

603 fan zr Page 25

20/9/2019 7:40

604 fan zr Page 25

20/9/2019 7:40

605 fan zr Page 25

20/9/2019 7:40

606 fan zr Page 25

20/9/2019 7:40

607 fan zr Page 25

20/9/2019 7:41

608 fan zr Page 25

20/9/2019 7:41

609 fan zr Page 25

20/9/2019 7:42

610 fan zr Page 25

20/9/2019 7:42

20:
2016. Wide & deep learning for recommender systems. In Recsys.
27:
2016. Deep neural networks for youtube recommendations. In Recsys.
129:
2016. Deep Crossing: Web-scale modeling without manually
crafted combinatorial features. In SIGKDD.

611 fan zr Page 25

20/9/2019 7:50

推荐研究领域也没有对特征工程有足够的研究

612 fan zr Page 25

20/9/2019 7:44

613 fan zr Page 25

20/9/2019 7:45

614 fan zr Page 25

20/9/2019 7:51

一个前瞻方向是设计最能探索其他模式数据的深度架构

615 fan zr Page 25

20/9/2019 7:45

616 fan zr Page 25

20/9/2019 7:50

197：

2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In CIKM.

617 fan zr Page 25

20/9/2019 7:50

618 fan zr Page 25

20/9/2019 7:51

619 fan zr Page 25

20/9/2019 7:52

620 fan zr Page 25

20/9/2019 7:52

621 fan zr Page 25

20/9/2019 7:52

622 fan zr Page 26

20/9/2019 8:03

623 fan zr Page 26

20/9/2019 8:04

624 fan zr Page 26

20/9/2019 8:04

625 fan zr Page 26

20/9/2019 15:09

626 fan zr

20/9/2019 15:10

627 fan zr

20/9/2019 15:10

126：

2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Recsys.

178：

2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks.

145：

2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In WWW.

628 fan zr

20/9/2019 15:10

629 fan zr

20/9/2019 15:11

630 fan zr

20/9/2019 15:11

146：

2018. Multi-Pointer Co-Attention Networks for Recommendation. In SIGKDD.

631 fan zr

20/9/2019 15:43

632 fan zr

20/9/2019 15:43

633 fan zr

20/9/2019 15:44

18：

2018. Visually Explainable Recommendation.

634 fan zr Page 26

20/9/2019 15:45

87：

2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation.

635 fan zr Page 26

20/9/2019 15:44

636 fan zr Page 26

20/9/2019 15:45

637 fan zr Page 26

20/9/2019 15:45

53：

2017. Neural collaborative filtering. In WWW.

177：

2016. Collaborative denoising auto-encoders for top-n recommender systems. In WSDM.

195：

2018. NeuRec: On Nonlinear Transformation for Personalized Ranking.

48：

2016. Deep residual learning for image recognition. In Proceedings ofthe IEEE conference on computer vision and pattern recognition.

64：

2017. Densely Connected Convolutional Networks.. In

147：

2018. Learning longer-term dependencies in rnns with auxiliary losses.

638 fan zr Page 26

21/9/2019 0:23

639 fan zr Page 26

21/9/2019 0:23

640 fan zr Page 26

21/9/2019 0:23

641 fan zr Page 26

21/9/2019 0:41

642 fan zr Page 26

21/9/2019 0:40

643 fan zr Page 26

21/9/2019 0:42

67：

2018. Compositional attention networks for machine reasoning.

124：

2017. A

simple neural network module for relational reasoning. In NIPS.

181：

2016. Dynamic coattention networks for question answering.

644 fan zr Page 26

21/9/2019 0:41

645 fan zr Page 26

21/9/2019 0:41

646 fan zr Page 26

21/9/2019 0:41

647 fan zr Page 26

21/9/2019 0:41

648 fan zr Page 26

21/9/2019 0:45

649 fan zr Page 26

21/9/2019 1:05

650 fan zr Page 26

21/9/2019 0:46
155 :
2017.
Attention is all you need. In Advances in Neural Information Processing Systems.
181 :
2016. Dynamic coattention networks for question answering.
146 :
2018. Multi-Pointer Co-Attention Networks for Recommendation. In SIGKDD.
62 :
2018. Leveraging Meta-path based Context for Top-N Recommendation with
A Neural Co-Attention Model. In SIGKDD.

651 fan zr Page 27

21/9/2019 1:06

652 fan zr Page 27

21/9/2019 1:07

653 fan zr Page 27

21/9/2019 1:07

654 fan zr Page 27

21/9/2019 1:07

655 fan zr Page 27

21/9/2019 1:10
74 :
2017. Cross Domain Recommender Systems: A Systematic Literature
Review.
40 :
2012. Cross-domain recommender systems: A survey
of the state of the art.
115 :
2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction.
In AAAI,
39 :
2015. A multi-view deep learning approach for cross domain user modeling in
recommendation systems. In WWW.
92 :
2017. CCCFNet: A Content-Boosted Collaborative Filtering Neural
Network for Cross Domain Recommender Systems. In WWW.

656 fan zr Page 27

21/9/2019 1:12

657 fan zr Page 27

21/9/2019 1:13

658 fan zr Page 27

21/9/2019 1:19

659 fan zr Page 27

21/9/2019 1:21

660 fan zr Page 27

21/9/2019 1:21

661 fan zr Page 27

21/9/2019 1:22

662 fan zr Page 27

21/9/2019 1:21

26：
2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings ofthe 25th international conference on Machine learning.
31：
2014. Deep learning: methods and applications.
5：
2016. Ask the gru: Multi-task learning for deep text recommendations. In Proceedings ofthe 10th ACM Conference on Recommender Systems.
73：
2017. Neural survival recommender. In WSDM.
87：
2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation.
187：
2016. Expanded autoencoder recommendation framework and its application in movie recommendation. In SKIMA.
111：
2010. Multi-task learning for recommender system. In Proceedings of2nd Asian Conference on Machine
Learning.

663 fan zr Page 27

21/9/2019 1:22

664 fan zr Page 27

21/9/2019 1:24

665 fan zr Page 27

21/9/2019 1:24

666 fan zr Page 27

21/9/2019 1:25

667 fan zr Page 27

21/9/2019 1:25

668 fan zr Page 27

21/9/2019 1:25

669 fan zr Page 27

21/9/2019 1:41

670 fan zr Page 27

21/9/2019 1:41

671 fan zr Page 27

21/9/2019 1:43

672 fan zr Page 27

21/9/2019 1:44

673 fan zr Page 27

21/9/2019 1:44

674 fan zr Page 27

21/9/2019 1:44

675 fan zr Page 27

21/9/2019 1:44

676 fan zr Page 27

21/9/2019 1:44

677 fan zr Page 27

21/9/2019 1:45

678 fan zr Page 27

21/9/2019 1:45

679 fan zr Page 27

21/9/2019 1:45

680 fan zr Page 27

21/9/2019 1:26
109：
2015. Deep
learning applications and challenges in big data analytics. Journal ofBig Data
144：
2018. Ranking Distillation: Learning Compact Ranking Models With High Performance for Recommender
System. In SIGKDD.
128：
2017. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In Recsys.

681 fan zr Page 28

21/9/2019 1:45

682 fan zr Page 28

21/9/2019 1:45

683 fan zr Page 28

21/9/2019 1:46

684 fan zr Page 28

21/9/2019 1:46

685 fan zr Page 28

21/9/2019 1:54

686 fan zr Page 28

21/9/2019 1:54

687 fan zr Page 28

21/9/2019 1:56

201 :
2018. MARS: Memory Attention-Aware
Recommender System.
53 :
2017. Neural collaborative filtering. In WWW.

688 fan zr Page 28

21/9/2019 1:56

689 fan zr Page 28

21/9/2019 1:57

690 fan zr Page 28

21/9/2019 1:58

691 fan zr Page 28

21/9/2019 1:58

692 fan zr Page 28

21/9/2019 1:58

693 fan zr Page 28

21/9/2019 1:58

694 fan zr Page 28

21/9/2019 1:59

695 fan zr Page 28

21/9/2019 1:59

696 fan zr Page 28

21/9/2019 2:00

697 fan zr Page 28

21/9/2019 2:01

698 fan zr Page 28

12/9/2019 4:13

699 fan zr Page 28

14/9/2019 1:47

700 fan zr Page 28

19/9/2019 9:22

701 fan zr Page 28

18/9/2019 15:56

702 fan zr Page 29

14/9/2019 2:18

703 fan zr Page 29

12/9/2019 1:58

704 fan zr Page 29

20/9/2019 3:14

705 fan zr Page 29

17/9/2019 13:56

706 fan zr Page 29

14/9/2019 1:51

707 fan zr Page 29

14/9/2019 1:47

708 fan zr Page 29

19/9/2019 9:22

709 fan zr Page 29

12/9/2019 7:48

710 fan zr Page 29

20/9/2019 6:55

711 fan zr Page 29

20/9/2019 15:44

712 fan zr Page 29

11/9/2019 13:58

713 fan zr Page 29

20/9/2019 1:16

714 fan zr Page 29

19/9/2019 9:22

715 fan zr Page 29

19/9/2019 9:22

716 fan zr Page 29

18/9/2019 15:28

717 fan zr Page 29

14/9/2019 2:15

718 fan zr Page 29

21/9/2019 1:22

719 fan zr Page 29

14/9/2019 1:48

720 fan zr Page 29

18/9/2019 15:43

721 fan zr Page 29

12/9/2019 7:12

722 fan zr Page 30

18/9/2019 15:29

723 fan zr Page 30

19/9/2019 15:13

724 fan zr Page 30

14/9/2019 16:39

725 fan zr Page 30

14/9/2019 1:48

726 fan zr Page 30

14/9/2019 1:51

727 fan zr Page 30

21/9/2019 1:11

728 fan zr Page 30

19/9/2019 1:57

729 fan zr Page 30

13/9/2019 3:35

730 fan zr Page 30

12/9/2019 7:48

731 fan zr Page 30

12/9/2019 7:56

732 fan zr Page 30

14/9/2019 1:48

733 fan zr Page 30

14/9/2019 1:48

734 fan zr Page 30

21/9/2019 0:21

735 fan zr Page 30

14/9/2019 2:15

736 fan zr Page 30

14/9/2019 2:15

737 fan zr Page 30

14/9/2019 2:19

738 fan zr Page 30

20/9/2019 3:14

739 fan zr Page 30

13/9/2019 4:04

740 fan zr Page 30

13/9/2019 4:11

741 fan zr Page 30

18/9/2019 1:54

742 fan zr Page 30

12/9/2019 8:35

743 fan zr Page 30

18/9/2019 6:30

744 fan zr Page 30

13/9/2019 4:20

745 fan zr Page 30

13/9/2019 4:21

746 fan zr Page 30

13/9/2019 6:24

747 fan zr Page 30

20/9/2019 7:34

748 fan zr Page 30

19/9/2019 14:49

749 fan zr Page 30

17/9/2019 9:09

750 fan zr Page 30

21/9/2019 0:21

751 fan zr Page 30

17/9/2019 1:46

752 fan zr Page 31

14/9/2019 1:49

753 fan zr Page 31

21/9/2019 0:42

754 fan zr Page 31

18/9/2019 6:38

755 fan zr Page 31

12/9/2019 6:06

756 fan zr Page 31

19/9/2019 11:04

757 fan zr Page 31

18/9/2019 15:28

758 fan zr Page 31

12/9/2019 1:59

759 fan zr Page 31

14/9/2019 2:16

760 fan zr Page 31

14/9/2019 2:16

761 fan zr Page 31

18/9/2019 1:35

762 fan zr Page 31

17/9/2019 11:54

763 fan zr Page 31

17/9/2019 11:57

764 fan zr Page 31

12/9/2019 7:55

765 fan zr Page 31

20/9/2019 6:54

766 fan zr Page 31

14/9/2019 2:19

767 fan zr Page 31

17/9/2019 13:48

768 fan zr Page 31

20/9/2019 7:30

769 fan zr Page 31

20/9/2019 7:30

770 fan zr Page 31

20/9/2019 1:20

771 fan zr Page 31

18/9/2019 15:27

772 fan zr Page 31

17/9/2019 9:29

773 fan zr Page 31

17/9/2019 9:11

774 fan zr Page 31

19/9/2019 11:08

775 fan zr Page 31

18/9/2019 15:28

776 fan zr Page 31

14/9/2019 1:49

777 fan zr Page 31

14/9/2019 1:51

778 fan zr Page 31

17/9/2019 6:28

779 fan zr Page 31

14/9/2019 1:50

780 fan zr Page 31

20/9/2019 3:19

781 fan zr Page 31

14/9/2019 2:16

782 fan zr Page 32

19/9/2019 11:04

783 fan zr Page 32

19/9/2019 1:58

784 fan zr Page 32

19/9/2019 11:08

785 fan zr Page 32

18/9/2019 6:39

786 fan zr Page 32

19/9/2019 9:23

787 fan zr Page 32

14/9/2019 2:16

788 fan zr Page 32

12/9/2019 7:59

789 fan zr Page 32

20/9/2019 1:17

790 fan zr Page 32

21/9/2019 1:42

791 fan zr Page 32

14/9/2019 2:19

792 fan zr Page 32

21/9/2019 1:23

793 fan zr Page 32

14/9/2019 1:52

794 fan zr Page 32

18/9/2019 15:58

795 fan zr Page 32

17/9/2019 7:18

796 fan zr Page 32

21/9/2019 1:10

797 fan zr Page 32

18/9/2019 2:00

798 fan zr Page 32

17/9/2019 13:49

799 fan zr Page 32

15/9/2019 14:31

800 fan zr Page 32

17/9/2019 9:27

801 fan zr Page 32

12/9/2019 4:13

802 fan zr Page 32

17/9/2019 11:54

803 fan zr Page 32

19/9/2019 1:47

804 fan zr Page 32

21/9/2019 0:42

805 fan zr Page 32

17/9/2019 3:31

806 fan zr Page 32

13/9/2019 6:20

807 fan zr Page 33

14/9/2019 2:17

808 fan zr Page 33

21/9/2019 1:42

809 fan zr Page 33

20/9/2019 7:43

810 fan zr Page 33

14/9/2019 2:17

811 fan zr Page 33

14/9/2019 2:23

812 fan zr Page 33

18/9/2019 6:30

813 fan zr Page 33

14/9/2019 1:52

814 fan zr Page 33

18/9/2019 16:03

815 fan zr Page 33

17/9/2019 6:11

816 fan zr Page 33

17/9/2019 6:12

817 fan zr Page 33

18/9/2019 1:42

818 fan zr Page 33

12/9/2019 8:35

819 fan zr Page 33

21/9/2019 1:43

820 fan zr Page 33

13/9/2019 4:05

821 fan zr Page 33

13/9/2019 6:21

822 fan zr Page 33

21/9/2019 0:22

823 fan zr Page 33

21/9/2019 0:22

824 fan zr Page 33

14/9/2019 2:20

825 fan zr Page 33

20/9/2019 7:33

826 fan zr Page 33

20/9/2019 7:31

827 fan zr Page 33

20/9/2019 7:31

828 fan zr Page 33

12/9/2019 7:55

829 fan zr Page 33

14/9/2019 2:17

830 fan zr Page 33

14/9/2019 1:52

831 fan zr Page 33

19/9/2019 9:23

832 fan zr Page 34

14/9/2019 1:50

833 fan zr Page 34

17/9/2019 9:10

834 fan zr Page 34

17/9/2019 8:20

835 fan zr Page 34

20/9/2019 7:02

836 fan zr Page 34

17/9/2019 8:21

837 fan zr Page 34

20/9/2019 1:47

838 fan zr Page 34

20/9/2019 7:22

839 fan zr Page 34

20/9/2019 3:15

840 fan zr Page 34

14/9/2019 2:17

841 fan zr Page 34

14/9/2019 1:50

842 fan zr Page 34

17/9/2019 9:30

843 fan zr Page 34

20/9/2019 1:17

844 fan zr Page 34

14/9/2019 2:20

845 fan zr Page 34

17/9/2019 11:57

846 fan zr Page 34

17/9/2019 11:57

847 fan zr Page 34

14/9/2019 2:18

848 fan zr Page 34

18/9/2019 15:27

849 fan zr Page 34

12/9/2019 8:35

850 fan zr Page 34

18/9/2019 1:42

851 fan zr Page 34

17/9/2019 6:15

852 fan zr Page 34

13/9/2019 6:21

853 fan zr Page 34

21/9/2019 0:43

854 fan zr Page 34

14/9/2019 1:52

855 fan zr Page 34

14/9/2019 1:52

856 fan zr Page 35

15/9/2019 14:30

857 fan zr Page 35

14/9/2019 1:50

858 fan zr Page 35

20/9/2019 7:33

859 fan zr Page 35

21/9/2019 1:24

860 fan zr Page 35

17/9/2019 9:12

861 fan zr Page 35

19/9/2019 11:04

862 fan zr Page 35

14/9/2019 2:20

863 fan zr Page 35

14/9/2019 2:20

864 fan zr Page 35

20/9/2019 3:18

865 fan zr Page 35

19/9/2019 14:42

866 fan zr Page 35

19/9/2019 14:39

867 fan zr Page 35

13/9/2019 4:10

868 fan zr Page 35

17/9/2019 11:54

869 fan zr Page 35

13/9/2019 3:39

870 fan zr Page 35

20/9/2019 1:18

871 fan zr Page 35

20/9/2019 1:18

872 fan zr Page 35

20/9/2019 1:18

873 fan zr Page 35

21/9/2019 1:56

874 fan zr Page 35

13/9/2019 3:34

875 fan zr Page 35

19/9/2019 15:13

876 fan zr Page 35

19/9/2019 14:53

877 fan zr Page 35

19/9/2019 14:41

878 fan zr Page 35

14/9/2019 2:18