A SURVEY OF DEEP LEARNING BASED RECOMMENDER SYSTEM

Wencao Yang
Strike Exchange LLC
yangwencao@gmail.com

ABSTRACT

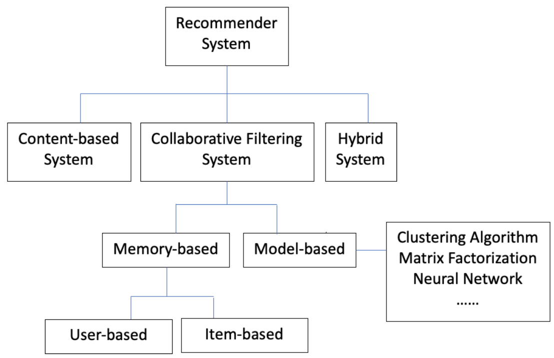
Recommender systems are used in various areas like content consumption and product recommendation. In recent years, deep learning has gained significant success in natural language processing and computational vision and has been applied in the field of recommender systems. In this survey, we give a review of the recent development of deep learning based recommender systems. We introduce the development of deep learning based recommender systems and describe the typical deep learning methods widely used in recommender systems. Some architecture of different models is present and compared.

Keywords Recommender System · Neural Network · Collaborative Filtering · Embedding · Knowledge Graph

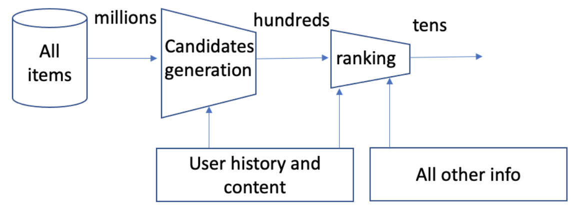
1 Introduction

Given the exploding growth of information on the Internet and access to overwhelming choices, recommender systems have been indispensable tools to retrieve and rank items for different users. Recommender systems have been widely used in content consumption (movie, music, news, ...) and E-commerce. It usually utilizes user-item interaction and side information about items or users to capture users' short- and/or long-term interest and then make predictions or recommendations. With the recent development of machine learning, especially deep learning, a significant amount of progress has been made to improve the quality of recommender systems, facing challenges such as accuracy, sparsity, cold-start, scalability, etc. And many advanced recommendation algorithms have been developed and deployed by platforms such as Youtube [1], Netflix [2], eBay [3], TikTok [4] etc.
However, there was a lack of extensive review on deep learning based recommender systems until S. Zhang et al. [5], and Z. Batmaz et al. [6] published their comprehensive, in-depth reviews. Short surveys are good for beginners in the field compared with comprehensive reviews. Several short surveys [7, 8] have been formally published. However, they failed to cover specific topics like knowledge graphs, sequence embedding, and attention-based models. Thus, this survey will focus on important progress and recent developments, especially the knowledge graph. Hopefully, this survey can provide some information and insight about recommender systems. This survey is organized as follows: Section 1 is a brief introduction to the recommender system from a high-level perspective. Section 2 briefly describes the recent development of the recommender system. Section 3 introduces the basic concepts and terminologies of deep learning techniques used in recommender systems. Section 4 discusses the usage of knowledge graphs in recommender systems. Section 5 puts together the architecture of different recommender system models and makes a comparison. Finally, Section 6 gives a conclusion and expectation of future work.

As shown in Figure 1(a), recommender systems can be classified into three categories [9]: collaborative filtering (CF) systems, content-based systems, and hybrid systems. The collaborative filtering systems utilize the interaction of users and items to find related or similar users and items. They can be further classified into memory-based and model-based methods. The memory-based method is to identify similar users (user-based) or items (item-based) based on the user-item matrix, which usually suffers from sparse data issues. The model-based method applies machine learning models like matrix factorization or neural networks to predict the rating of a new item. Content-based systems exploit side information like text, visual (picture), and audio descriptions of items. Hybrid systems are a combination of the above and utilize multiple approaches.
Figure 1(b) shows a general structure of a recommender system. We divide the recommender system into retrieval and ranking stages due to the vast number of items.



(a) Classification of recommender systems



(b) Structure of a recommender system

Figure 1: Summary of recommender systems

2 Development of Recommender System

Content-based filtering uses item features to recommend other similar items based on the user's previous actions, explicit or implicit. Collaborative filtering uses similarities between users and items simultaneously. For instance, to predict the ratings of certain items for one user, we look for users who have similar tastes and use the ratings from these users to predict.
Netflix Prize competitions [2], began on Oct, 2006, sparked interest in matrix factorization (MF) [10], due to its superior performance compared with classic nearest-neighbor techniques. MF can be considered as a simple embedding model. Figure 2 gives an example for MF, where the $m \times n$ user-item matrix decomposes into $m \times d$ user embedding matrix and $n \times d$ item embedding matrix ($d = 2$ in this example). Factorization machine (FM) [11] is a generalization of the linear regression and matrix factorization models. FM captures the linear and pairwise interactions between features where the interaction is the dot-product of features. Moreover, the so-called neural collaborative filtering extends the model capacity by learning the highly non-linear interaction with the neural network.

Figure 2: An example of MF. Image taken from [12]

Compared with traditional matrix factorization, which decomposes the user-item matrix to obtain user and item latent vectors, neural collaborative filtering models utilize the neural networks to construct the user and item latent vector. The commonly used networks are multiple layer perception (MLP), convolutional neural network (CNN), recurrent neural network (RNN), long-short-term-memory (LSTM), gated recurrent unit (GRU), etc. As illustrated in Figure 3, many models have been proposed, including Deep Crossing [13], Factorisation-machine supported Neural Networks (FNN) [14], Product-based Neural Network (PNN) [15], Wide & Deep Model [16], DeepFM [17], Deep & Cross Network (DCN) [18], Attentional Factorization Machines (AFM) [19], Deep Interest Network (DIN) [20], and Deep Interest Evolution Network (DIEN) [21]. Further discussion can be found in Section 5.

deep neural networks for youtube recommendations [1] is a classic paper showing how the neural network is used in video retrieval and ranking in the real world. In this paper, video retrieval is considered an extreme multiclass classification problem. As shown in Figure 4, embedded video watches and search tokens are averaged and concatenated

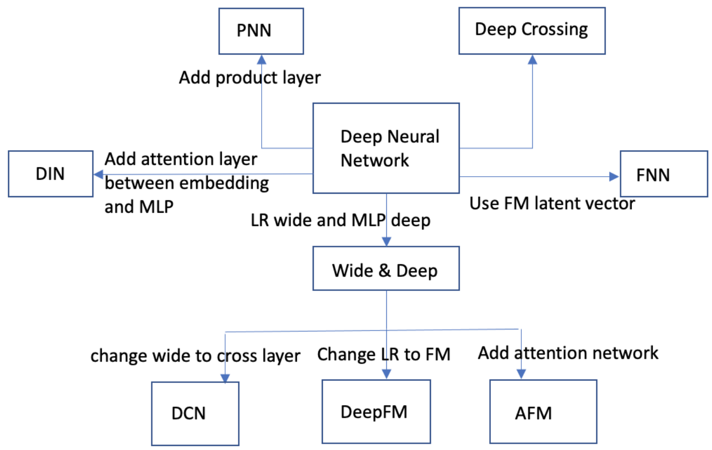


Figure 3: An overview of deep learning models.

with other demographic embeddings and fed into a fully connected neural network. A similar neural network is used for ranking. The main difference is the number of input features, much more features are used for ranking, such as *time since last watch*, *number of previous impressions*.

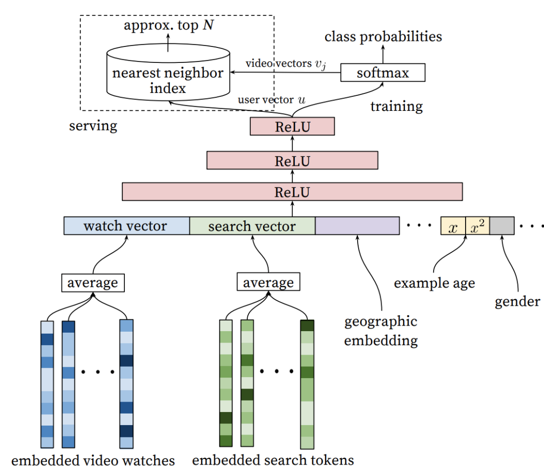


Figure 4: An overview of DNN for youtube. Image taken from [1]

The transformer has been the start-of-art feature extractor since it was proposed in the field of natural language processing (NLP). It has achieved start-of-art performance in various tasks, and it is natural to use attention networks in recommender systems. Many recommender system models with attention mechanism have been proposed, including DeepCoNN [22], TransNet [23], D-ATT [24]. A detailed discussion is out of the scope of this survey.

Recently, much effort has been made to integrate knowledge graphs into recommender systems because knowledge graphs can enhance recommendation by providing side information. Embedding-based and path-based methods are developed to take advantage of knowledge graphs in recommendation tasks. A detailed discussion can be found in Section 4.

3 Techniques of Deep Learning

This section describes the basic concepts of some deep learning technologies and their application in recommender systems.

3.1 Autoencoder

An autoencoder is a type of neural network attempting to reconstruct its input data in the output layer. For recommendation, autoencoders are mainly used in learning low-dimensional representation (embedding) of features at the hidden layer or predicting the missing value at the reconstruction layers for purposes of recommendations. AutoRec [25], CDAE [26] are examples of recommender system using autoencoders.

3.2 Convolutional Neural Network

Convolutional neural networks (CNN)[27, 28] is a class of neural networks widely used in the field of computer vision (CV). As shown in Figure 5, a typical CNN is usually composed of convolutional layers, pooling (subsampling) layers, and fully connected layers. A CNN can capture global and local information and is powerful in extracting features, making it attractive in dealing with the image or textual data.

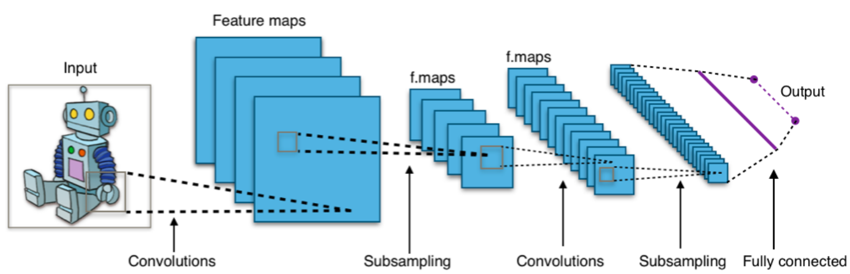


Figure 5: A typical CNN architecture. Image taken from [29]

CNN has been used in fashion-trend [30, 31], and style preference [32] where visual appearance is considered for the personal ranking. CNN is also used for point-of-interest (POI) recommendation by exploiting images [33], and document encoding for news recommendation [34]. ConvMF [35] is proposed to integrate CNN into probabilistic matrix factorization (PMF), where the CNN generates latent vectors for documents. J. McAuley et al. [36] applied CNN to uncover relationships between the appearances of pairs of objects. There are hybrid models such as attention-based CNNs [37] proposed for review rating prediction.

3.3 Recurrent Neural Network

A recurrent neural network (RNN) [38] is a type of neural network where connections between nodes form a directed or undirected graph along a temporal sequence. Previous values of the hidden units are saved and passed into the hidden units along with input values. As shown in Figure 6, the hidden layer h_t at time t depends not only input vector x_t, but also hidden layer h_{t-1} at time $t-1$

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{1}$$

Equation 1 shows how the hidden value is calculated, where σ_h is the activation fuction,W,U and b are parameter matrices and vector.

Variants such as long short-term memory (LSTM) [40], gated recurrent unit (GRU) [41] are also proposed to overcome the gradient vanishing issue.
RNN, LSTM, and GRU are suitable for time-aware or time-dependent recommender systems. In recommender systems, the recommendation may depend on the time, and the user and item features may influence each other, and both evolve. H. Dai et al. [42] proposed a novel deep coevolutionary network model (DeepCoevolve), which uses RNN to learn user and item features based on time-dependent user-item interaction graph. T. Bansal et al.[43] proposed architecture with bi-directional RNN and GRU for text item recommendations. T. Donkers et al. [44] proposed a user-based GRU for a sequential recommendation. B. Hidasi et al. [45] report the use of RNN in a session-based recommendation.

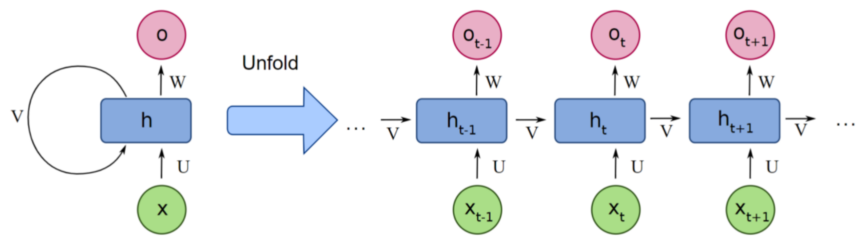


Figure 6: A RNN structure unfold by time. Image taken from [39]

3.4 Sequence Embedding

Sequence Embedding is not an architecture but an approach to get embedding for entities, becoming popular after word2vec [46]. word2vec is a technique for natural language processing (NLP). It uses a neural network to learn word association and has been a powerful tool since published in 2013. A detailed explanation of word2vec can be found in [47]. It uses continuous bag-of-word (CBOW) or continuous skip-gram for model training. In principle, any meaningful sequence can utilize a similar approach to generate embedding vectors for items. Users' history of behaviors (view/click/book/purchase) is a perfect example of coherent sequences and thus can be used for generating item embedding. item2vec [48] is a extension of word2vec. It ignores the spatial information and treats each pair of items in the same set as a positive example, and the objective function is as follow:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{j\neq i}^{K}\log p(w_j|w_i) \tag{2}$$

Airbnb has used a skip-gram model to learn the listing embeddings for the task of listing recommendations/ranking in search at Airbnb [49]. Click sessions were obtained from users, where each session is an uninterrupted sequence of listing ids were clicked by the user, and the skip-gram model is used to learn listing representations.

3.5 Attention-based methods

Transformer [50] has been a powerful feature extractor used in NLP models such as Bert [51], GPT-3 [52]. The key idea is the attention function. An attention function maps a query and a set of key-value pairs to an output, and the output is the weighted sum of values.

$$Attention(Q;K;V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

In equation 3, Q represent queries, K the keys and V the values. Self-attention based sequential model (SASRec) [53] is an example showing how an attention mechanism is used for sequential recommendation problems. As shown in Figure 7, at each time step, the input is the previous items, and the output is the next item.

Q. Zhang et al. [54] used attention mechanism in hashtag recommendation for multimodal microblog. Y. Tay et al. [55] proposed a Multi-Pointer Co-Attention Networks and claimed it outperformed DeepCoNN [22], TransNet [23], and D-ATT [24].

4 Knowledge Graph

A knowledge graph sometimes is referred to as a knowledge base. A knowledge graph is a graph-structured representation of knowledge, a collection of interlinked entities, i.e., event, concept, or product. Knowledge graph embedding methods have been developed with the help of sequence embedding. Both DeepWalk [56] and node2vec [57] use random walks to get sequences from the knowledge graph and then use the sequence embedding method to get entity embedding.
Another approach represents relationships between entities as translations in the embedding space. Relationship between entities is usually written as a triplet (h, r, t), where h is the head entity, t is the tail entity and r is the relationship. Models using this approach such as TransE [58], TransH [59], TransR [60], TransD [61] have been

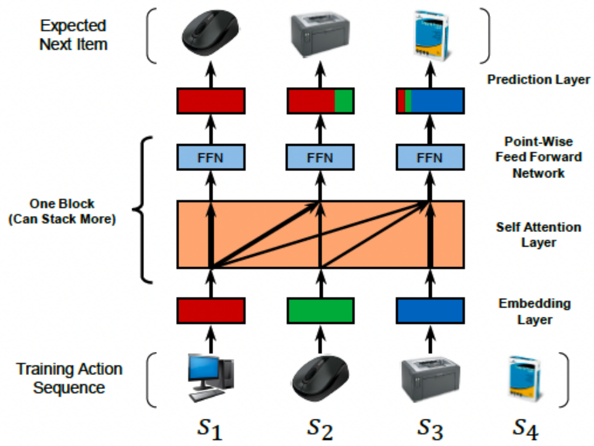


Figure 7: SASRec. Image taken from [53]

proposed to embed knowledge graphs. As illustrated in Figure 8, TransR mapps entity embeddings from entity space to relation space, and the difference between h_r and t_r is the relationship r.

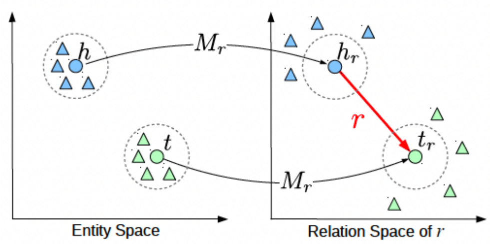


Figure 8: Simple illustration of TransR. Image taken from [60]

Knowledge graphs can enhance the recommender system in three ways: (1) Knowledge graphs introduce semantic relatedness among items; (2) Knowledge graphs contain different types of relationships, which may extend users' interests and increase the diversity of recommended items; (3) Knowledge graphs connect users' historical records and the recommended ones, improving explainability. Various models have been proposed to integrate knowledge graphs into the recommender system. They can be classified into three categories [62]: 1. Embedding-based methods 2. Path-based methods 3. Unified methods.

4.1 Embedding-based methods

The embedding-based methods use the embeddings learned from the knowledge graph directly to enrich the representation of users or items. For instance, Collaborative Knowledge Based (CKE) model [63] uses Bayesian transR, Bayesian stacked denoising auto-encoders (SDAE), and Bayesian stacked convolutional auto-encoders (SCAE) to obtain structural embedding, textual embedding and visual embedding, and add up these embedding with the latent item offset vector.
Deep Knowledge-Aware Network (DKN) [34] was proposed in news recommendation, where news title embeddings are learned with Kim CNN [64]. Words from news titles are concatenated with corresponding entities learned by TransD and then fed into the CNN. The attention mechanism is also used in the model.
Other works include entity2rec [65], SHINE [66], RCF [67], etc.
The works above directly used the raw latent vectors learned from knowledge graph embedding. Some work has been proposed to improve the recommendation performance by refining the entity embeddings.

4.2 Path-based methods

The knowledge graph is often named heterogeneous information networks (HIN) in path-based models. Path-based methods use the connectivity patterns of entities. Connectivity similarity is learned and used to enhance recommendation. PathSim [68] is commonly used to measure the similarity. For example, Personalized Entity Recommendation (PER) [69] model defines meta-path $\mathcal{P}$ and evaluate the score between user i and item j along path $\mathcal{P}$ with PathSim.
PER model needs to pre-define meta path $\mathcal{P}$. Models without pre-defined meta-paths have been proposed. One example is Knowledge-aware Path Recurrent Network (KPRN) [70]: *KPRN takes a set of paths of each user-item pair as input, and outputs a score indicating how possible the user will interact the target item.* This can be seen in Figure 9. The input embedding layers contain three layers: entity, entity type, and relation type. After fed into the LSTM layers, it's expected to predict the interaction between the start entity (Alice, e_1) and the end entity (I see Fire, e_5).

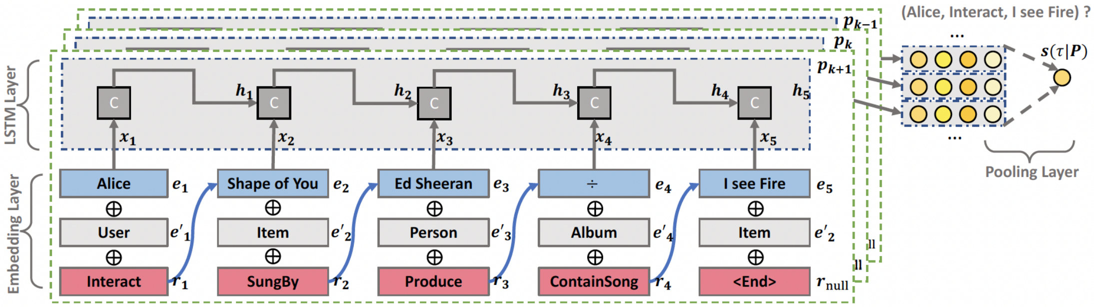


Figure 9: Schematic overview of KPRN. Image taken from [70]

Besides KPRN, there are other models proposed, such as Recurrent Knowledge Graph Embedding (RKGE) [71], Meta-Graph based recommendation fusion over HIN [72], Meta-path based Recommendation with A Neural Co-Attention Model [73].
Path-based methods generally extract meta-path or automatically learn the path pattern to help enhance recommendations.

4.3 Unified methods

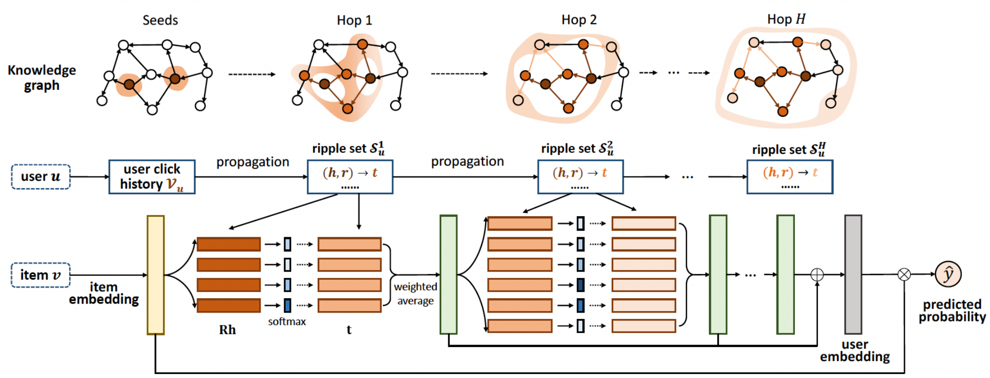
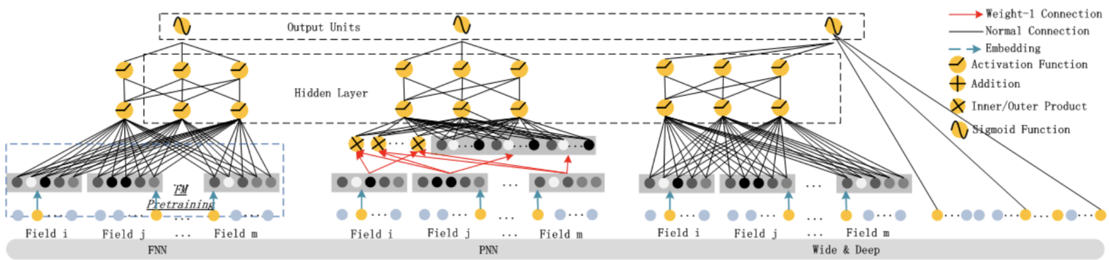
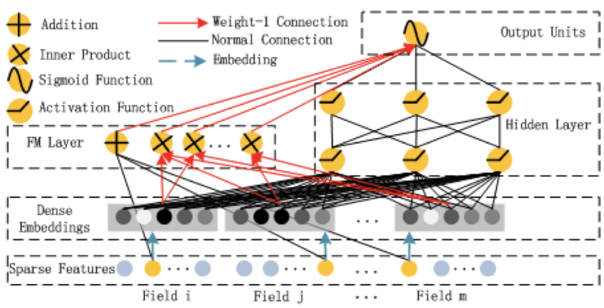


Figure 10: The overall framework of the RippleNet. Image taken from [74]

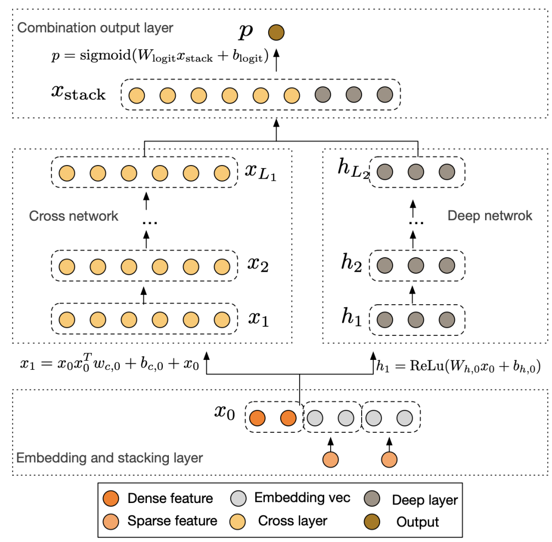
Unified methods benefit from both embedding-based and path-based methods.
RippleNet [74] is the first work to combine embedding-based and path-based methods in knowledge graph aware recommendation. In RippleNet, the user embedding is obtained by a user's history. The key idea behind RippleNet is *preference propagation*. As illustrated in Figure 10, in short, the user's clicked items are seed entities, and (h, r, t) is the corresponding $1 - hop$ ripple set, where h is in the seed entities, and t is in the $1 - hop$ entities, which is the h of $2 - hop$ ripple set. The $k - hop$ ripple set is obtained from $(k-1) - hop$ ripple set. Relevance probability P_i is calculated between item v and the head h_i of the $k - hop$ ripple set. Vector o^k is weighted sum of tail t_i, where the



(a) FNN, PNN, Deep & Wide, Image taken from [17]



(b) DeepFM. Image taken from [17]



(c) DCN, Image taken from [18]

Figure 11: Different Architectures

weight is P_i. The user embedding is the sum of all order vectors o^k.

Y. Cao et al. [75] proposed a model unifying knowledge graph learning and recommendation.

5 Architectures of Deep Learning based Recommender Systems

In this section, we directly compare some typical architectures.
The deep crossing model proposed by Microsoft in 2016 is the classic and base model of deep learning recommendation. It converts features into dense embeddings, concatenates them into a stacking layer, and feeds them into multiple residual units. Compared with Deep Crossing, FNN uses the latent vector learned from FM as the initial embeddings instead of random initialization. PNN adds a product layer between the embeddings and the hidden layers, where the product layer can be an inner product or an outer product. The Wide & Deep model connects a wide part (memorization) and a deep part (generalization) and feeds into the final sigmoid function. Figure 11(a) shows the difference between FNN, PNN, and Wide & Deep. As illustrated in Figure 11(b), the DeepFM replaces the wide part of the Wide % Deep model with an FM component, capturing order-2 features interaction more efficiently because it has inner product unites. Figure 11(c) shows that DCN is similar to DeepFM, and it replaces the wide part with a cross network to increase the feature interactions. The definition and calculation of cross layer can be in [18]. AFM [19] is nothing but FM with attention mechanism, DIN [20] applies attention mechanism in the neural network, and DIEN [21] learns the evolution of interest.

6 Conclusion and future work

S. Zhang et al. [5] have pointed out some future research directions and open issues, such as *Explainable Recommendation with Deep Learning*, *Cross Domain Recommendation with Deep Neural Networks*, *Deep Multi-Task Learning for Recommendation*, and *a suite of standardized evaluation datasets*. We encourage readers to explore this review to figure out more open issues.
The astronomical amount of data generated on the Internet poses enormous challenges for information seeking. Deep learning-based recommender systems (DLRS) are leading solutions to these challenges. Although far from comprehensive, we hope readers can get a general understanding of the recent progress in the recommender system from this survey.

References

[1] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA, 2016. Association for Computing Machinery.

[2] James Bennett, Stan Lanning, and Netflix Netflix. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.

[3] Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. Personalized embedding-based e-commerce recommendations at ebay. *CoRR*, abs/2102.06156, 2021.

[4] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. Jointly learning to recommend and advertise. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3319–3327, New York, NY, USA, 2020. Association for Computing Machinery.

[5] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1), feb 2019.

[6] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.*, 52(1):1–37, jun 2019.

[7] Ruihui Mu. A survey of recommender systems based on deep learning. *IEEE Access*, 6:69009–69022, 2018.

[8] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. Deep learning methods on recommender system: A survey of state-of-the-art. *International Journal of Computer Applications*, 162(10):17–22, Mar 2017.

[9] G. Adomavicius and A. Tuzhilin. 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):734–749, 2005.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[11] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000, 2010.

[12] Matrix factorization. `https://developers.google.com/machine-learning/recommendation/collaborative/matrix`.

[13] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 255–262, New York, NY, USA, 2016. Association for Computing Machinery.

[14] Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data: A case study on user response prediction, 2016.

[15] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Trans. Inf. Syst.*, 37(1), oct 2018.

[16] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, page 7–10, New York, NY, USA, 2016. Association for Computing Machinery.

[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1725–1731. AAAI Press, 2017.

[18] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions, 2017.

[19] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 3119–3125. AAAI Press, 2017.

[20] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction, 2017.

[21] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[22] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 425–434, New York, NY, USA, 2017. Association for Computing Machinery.

[23] Rose Catherine and William Cohen. Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 288–296, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 297–305, New York, NY, USA, 2017. Association for Computing Machinery.

[25] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 111–112, New York, NY, USA, 2015. Association for Computing Machinery.

[26] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, page 153–162, New York, NY, USA, 2016. Association for Computing Machinery.

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Convolutional neural network. `https://en.wikipedia.org/wiki/Convolutional_neural_network`.

[30] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[31] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 144–150. AAAI Press, 2016.

[32] Qiang Liu, Shu Wu, and Liang Wang. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 841–844, New York, NY, USA, 2017. Association for Computing Machinery.

[33] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 391–400, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[34] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1835–1844, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[35] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 233–240, New York, NY, USA, 2016. Association for Computing Machinery.

[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA, 2015. Association for Computing Machinery.

[37] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. 2017.

[38] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

[39] File:recurrent neural network unfold.svg. `https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg`.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[41] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[42] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation, 2016.

[43] Trapit Bansal, David Belanger, and Andrew McCallum. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 107–114, New York, NY, USA, 2016. Association for Computing Machinery.

[44] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 152–160, New York, NY, USA, 2017. Association for Computing Machinery.

[45] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2015.

[46] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[47] Xin Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014.

[48] Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering, 2016.

[49] Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 311–320, New York, NY, USA, 2018. Association for Computing Machinery.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[52] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[53] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation, 2018.

[54] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag recommendation for multimodal microblog using co-attention network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3420–3426, 2017.

[55] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Multi-pointer co-attention networks for recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, jul 2018.

[56] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[57] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[58] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[59] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014.

[60] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015.

[61] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China, July 2015. Association for Computational Linguistics.

[62] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems, 2020.

[63] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 353–362, New York, NY, USA, 2016. Association for Computing Machinery.

[64] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[65] Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 32–36, New York, NY, USA, 2017. Association for Computing Machinery.

[66] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 592–600, New York, NY, USA, 2018. Association for Computing Machinery.

[67] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 125–134, New York, NY, USA, 2019. Association for Computing Machinery.

[68] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proc. VLDB Endow.*, 4(11):992–1003, aug 2011.

[69] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, page 283–292, New York, NY, USA, 2014. Association for Computing Machinery.

[70] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[71] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 297–305, New York, NY, USA, 2018. Association for Computing Machinery.

[72] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 635–644, New York, NY, USA, 2017. Association for Computing Machinery.

[73] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. Leveraging meta-path based context for top- n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1531–1540, New York, NY, USA, 2018. Association for Computing Machinery.

[74] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 417–426, New York, NY, USA, 2018. Association for Computing Machinery.

[75] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference*. ACM, may 2019.