A Brief Review of Basic Deep Learning Models for Recommendation Systems

Xitong Zhou
Department of Earth Science and Engineering, Imperial College London, Exhibition Rd, South Kensington, London, U.K.

Keywords: Deep Learning, Recommendation System, Deep Neural Network.

Abstract: Recommendation systems are essential for delivering personalized content to users across various platforms, enhancing user experience and engagement. Traditional filtering methods, including content-based filtering and collaborative filtering, have been widely applied to recommend items based on user preferences or similarities between users and items. However, these methods still face challenges such as data sparsity, computational complexity, and the cold-start problem, which limit their effectiveness and scalability. This paper provides an overview of these traditional recommendation techniques, their limitations, and how deep learning approaches are transforming the field by addressing these issues. The discussion focuses on several deep learning models, including Multi-Layer Perceptrons (MLP), Autoencoders, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Generative Adversarial Networks (GAN). These models enhance recommendation systems by capturing complex, non-linear interactions between users and items, thereby significantly improving personalization, scalability, and robustness in cold-start scenarios. By leveraging the power of neural networks, deep learning is ushering in a new era for recommendation systems, offering more accurate, dynamic, and adaptive recommendations.

1 INTRODUCTION

The advent and widespread adoption of the internet have provided users with an abundance of information, meeting their needs in the information age. However, as the internet rapidly expands, the sheer volume of online information has grown substantially, which has made it increasingly difficult for users to identify the information that is truly useful to them amidst the vast quantities available, ultimately reducing the efficiency with which they can utilize this information.

Recommendation systems play an important role in improving user experience on diverse platforms by offering tailored suggestions. Over time, these systems have evolved from basic algorithms to more advanced data-driven approaches, enabling more accurate and relevant recommendations. Conventional recommendation systems, including content-based filtering and collaborative filtering, are prevalent yet encounter substantial obstacles, such as data sparsity, cold-start problems, and scalability limitations.

Deep learning leverages multi-layer deep neural networks to automatically learn from large datasets, making it particularly useful in recommendation systems. With increasing application of deep learning, new models have been continuously developed to overcome existing limitations. These models employ neural networks to replicate user-item interactions and to represent intricate patterns and nonlinear correlations in the data, providing more resilient solutions.

A Deep Neural Network (DNN) is an intricate artificial neural network architecture consisting of several layers of neurons. Each neuron is responsible for receiving input, processing it, and producing an output. The existence of several hidden layers positioned between the input and output layers is a DNN's defining feature (Samek, 2021).

- The **input layer** is the network's initial layer, responsible for accepting input data.
- The **hidden layers** form the core of the network, with each layer containing multiple neurons that process the data received from a previous layer and forward the results to the layer above.

Zhou and X.
A Brief Review of Basic Deep Learning Models for Recommendation Systems.
DOI: 10.5220/0013526300004619
In *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning (DAML 2024)*, pages 473-482
ISBN: 978-989-758-754-2

- The **output layer**, which is the network's last layer, has as many neurons as necessary to complete a certain job, like regression or classification, depending on its particular needs.

This multi-layer architecture allows DNNs to model complex relationships and extract high-level features from data, making them very effective for a variety of uses. A deep neural network framework for recommendation systems (Figure 1) can be illustrated with the following structure:

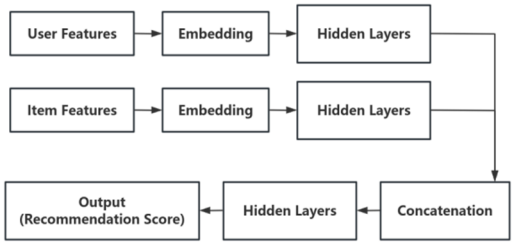

Figure 1: A General Architecture of DNNs in Recommendation Systems

This paper gives a brief summary of various types of recommendation systems and the difficulties that traditional approaches encounter. It then introduces how deep learning techniques, such as Multi-Layer Perceptron (MLP), Autoencoders, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Generative Adversarial Networks (GAN), have been applied to recommendation systems and how they improve upon traditional models. The paper concludes by outlining the problems and difficulties that deep learning models in recommendation systems are now facing and making recommendations for future study.

2 BACKGROUND

2.1 Types of Recommendation Systems

The primary types of recommendation systems include content-based, collaborative filtering, hybrid, social network-based, and knowledge graph-based approaches (Fayyaz, 2020). By examining a user's past behavior and personal data, content-based recommendation systems make recommendations relevant to their interests and preferences. For example, recommendations may include movies, music, or books that align with a user's tastes. Collaborative filtering systems match users or items with similar interests based on historical behavior, recommending content that similar users or items have interacted with.

Because of their unique advantages and disadvantages, each of these kinds is appropriate in certain situations. In practical applications, recommendation systems typically combine one or more methods depending on specific requirements to achieve better recommendation outcomes.

2.2 Challenges of Traditional Methods

The data sparsity, high processing complexity, and cold-start problem are the main problems that traditional recommendation systems confront. The restricted availability of user-item interaction data is referred to as data sparsity. Collecting enough rating data is challenging since consumers engage with recommendation systems in a variety of ways. For instance, users typically rate only items they like, while leaving many others unrated, resulting in sparse data.

Additionally, due to the long-tail effect, niche items can collectively make up a significant portion of the market. Therefore, it is particularly crucial for recommendation systems to provide a wide variety of niche or less popular items to the right users. However, in recommendation systems, a select few popular items tend to receive the majority of user ratings, while most niche items have very few ratings, making it challenging to model relationships between items. With more users and items added to the system, the computational complexity rises significantly. Traditional algorithms, such as matrix factorization, require substantial computational resources to handle large-scale data, making real-time recommendation extremely difficult.

Moreover, recommendation systems often encounter the cold-start issues. There are three primary forms of cold start issues: first, the lack of historical data for new users prevents the system from identifying their preferences, making personalized recommendations difficult; second, it is challenging to determine which consumers would be interested in new products uploaded to the system as there is insufficient engagement data available.; third, when the system is newly launched, the absence of sufficient interaction data between users and items makes it hard to provide any meaningful recommendations (Ko, 2022).

3 DEEP LEARNING MODELS IN RECOMMENDATION SYSTEMS

3.1 Multi-Layer Perceptron-based

Multi-Layer Perceptron (MLP) is an artificial neural network that simulates the connectivity of neurons in the human brain, enabling it to learn and extract complex features from massive datasets. By comparing the similarities between user preferences and item features, MLP may be used to build user profiles and provide accurate lists of recommendations.

The construction of user profiles involves collecting and analyzing behavioral data, personal information, and other relevant data to create personalized features for each user. These features may encompass various aspects such as interests, consumption habits, and geographical location. MLP can learn and model these features, allowing for more precise predictions of user preferences. Modeling item characteristics includes information about the content, attributes, and categories of items. By using MLP to learn and process item features, the system can uncover associations and similarities among items, ultimately providing recommendations aligned with user preferences (Zhou, 2022).

3.1.1 Neural Collaborative Filtering-based

The Neural Collaborative Filtering (Neural CF) model is an improvement over conventional collaborative filtering techniques (Matrix Factorization, MF) in that it makes use of a multi-layer neural network instead of using the dot product operation between the user and item vectors. This allows for more comprehensive interactions between the two vectors, yielding richer combinations of valuable feature information. To enable the neural network to fully perform collaborative filtering, a multi-layer perceptron is employed to mimic user-item interactions. The output from one layer is used as the input for the succeeding layer. Neural CF significantly improves the generalization and fitting capabilities of collaborative filtering algorithms (He, 2017).

To increase the model's functionality for both linear and nonlinear combinations, He et al. (2017) proposed a hybrid version of Neural Collaborative Filtering that integrates the original Neural CF model with a Multi-Layer Perceptron (MLP) and the element-wise product-based Generalized Matrix Factorization (GMF). In this model, the embedding vectors are learned separately rather than shared, providing greater flexibility—allowing for different dimensions of latent vectors to be determined based on the complexity of the model, and computed individually within their respective models. The results from these computations are then merged and passed through a further fully linked layer to generate the output (He, 2017).

The Joint Neural Collaborative Filtering (J-NCF) approach optimizes user-item ratings using a user-item rating matrix. This method combines deep feature learning with deep interaction modeling, enhancing recommendation performance by capturing nonlinear user-item interactions. The model's loss function considers pointwise and pairwise loss, as well as implicit and explicit feedback. J-NCF is skilled in scalability and sensitivity under varying data sparsity and user activity levels, especially addressing "inactive users" (Chen, 2019).

3.1.2 AutoMLP-based

Recently, Li proposed a long-short term sequence recommendation system named AutoMLP, designed to more accurately represent users' short-term and long-term interests in their past encounters. AutoMLP consists solely of Multi-Layer Perceptrons (MLPs), keeping complexity in time and space linear. Both long-term and short-term dependencies are captured by the model's long-short term interest module. Utilizing automation techniques, AutoMLP employs continuous relaxation to transform discrete sequence lengths into continuously differentiable representations, thereby adaptively optimizing the window of short-term interest for various tasks (Li, 2024).

3.2 Autoencoder-based

An autoencoder is a neural network model whose core idea is to use an **encoder** to convert input data into a low-dimensional feature representation, and a **decoder** to either produce new data or decode the old data back. Theoretically, an autoencoder can be considered a generative model, as it learns the data distribution and may produce fresh data samples. In recommendation systems, autoencoders are a useful tool for learning users' latent features to generate content that aligns with their preferences, thereby enabling more accurate personalized recommendations. From an application perspective, autoencoders can be employed to handle data in

recommendation systems, such as for dimensionality reduction and data compression.

3.2.1 Denoising Autoencoder-based

Denoising Autoencoders (DAEs) are a type of neural network model used for unsupervised learning (Vincent, 2008). Unlike standard autoencoders, which may simply replicate the input or extract trivial features, DAEs intentionally introduce noise into the input data and then attempt to recreate the initial, noise-free data. This approach enhances the constraints on the data, encouraging the model to acquire more practical feature representations.

By reconstructing the noisy input, DAEs achieve more robust feature representations, thus avoiding the issue of merely copying the original input. Overall, denoising autoencoders not only learn features similar to the original data but also demonstrate improved performance and stability in the presence of noise and uncertainty.

Stacked Denoising Autoencoders (SDAEs) were introduced by Vincent et al. as a deep network construction strategy, where these autoencoders are trained locally to denoise corrupted input versions (Vincent, 2010). Compared to standard autoencoders, this approach has shown a significant reduction in classification errors in benchmark tests. Building on the SDAE framework, Wang et al. proposed a probabilistic version of SDAE that integrates it with Probabilistic Matrix Factorization (PMF), leading to the development of the Relational SDAE (RSDAE) model. This model systematically combines deep and relational learning and can naturally extend to handle multi-relational data, effectively enhancing the performance and broad applicability of label recommendation tasks (Wang, 2015).

Wu et al. suggested the Collaborative Denoising Autoencoder (CDAE) for top-N recommendation. CDAE serves as a generalization of various existing collaborative filtering models, featuring a more flexible structure that effectively incorporates nonlinear components to enhance recommendation performance (Wu, 2016). Subsequently, Khan et al. proposed the User-Tracking Collaborative Denoising Autoencoder (UT-CDAE). This model evaluates user rating trends (high or low) across a group of items to determine user-item associations. The introduction of rating trends provides the model with additional regularization flexibility. By incorporating trend-based weighting, UT-CDAE is able to learn more robust and nonlinear latent representations, improving the ranking prediction capability of the output layer and thereby enhancing the accuracy of top-N recommendation predictions (Khan, 2019).

Considering an infinite number of copies of the training data that are corrupted, Chen et al. proposed the Marginal Denoising Autoencoder (MDAE), which addresses the issue of corruption by (approximately) marginalizing it during the training process (Chen, 2014). MDAE implicitly marginalizes all possible corrupted samples to reconstruct the error, thus avoiding additional computational costs. This enables MDAE to achieve or exceed the performance of traditional denoising autoencoders within fewer training iterations. Compared to other related works, MDAE not only supports nonlinear encoding and decoding but also excels in learning latent representations. Additionally, Marginalized Stacked Denoising Autoencoder (MSDAE) is a deep structure that can be created by stacking MDAE (Zhang, 2020).

3.2.2 Variational Autoencoder-based

Variational Autoencoders (VAEs) are a kind of neural network-based generative model that perform exceptionally well in unsupervised learning. Using random sampling, they produce new samples that are comparable to the original data but not exact replicas of it after discovering the latent distribution of the data samples (Kingma, 2013). This makes VAEs a promising approach for personalized recommendations in recommendation systems. To become familiar with the model parameters, VAEs maximize the marginal probability of the observed data during training.

The impact of VAEs on the development of recommendation systems is significant. They discover how users' past actions and interests relate to one another, generating new samples that align with user preferences but differ in specific ways. Additionally, VAEs can be used to discover hidden features and calculate similarities. By learning the similarities between users and applying this in recommendation systems, VAEs enable more accurate similarity computations by measuring distances in the latent space, thus enhancing the recommendation's accuracy. Furthermore, VAEs learn the relationships between users and items in the latent space, which is beneficial for addressing cold-start problems. By mapping new items or users into the latent space and comparing them with existing data, VAEs can provide personalized recommendations.

To further enhance collaborative filtering performance, Shenbin et al. (Shenbin, 2020) proposed

the Recommender VAE (RecVAE), based on VAE and Mult-VAE (Zhao, 2017). RecVAE presents a composite prior combining standard Gaussian priors with the latent code distribution from the previous iteration, improving training stability and performance. Additionally, RecVAE employs an alternating update training method, allowing the more complex encoder to be updated multiple times during each decoder update, using corrupted inputs for encoder training while the decoder uses clean inputs. These innovations significantly enhance the model's recommendation performance under implicit feedback (Shenbin, 2020).

Zhu et al. presented the Mutually-regularized Dual Collaborative Variational Autoencoder (MD-CVAE) to address sparsity and cold-start problems in collaborative filtering by using stacked latent item embeddings in place of the standard User Autoencoder (UAE)'s randomly initialized weights in the final layer, integrating user ratings with item content information within a unified variational framework. This prevents the model from converging to suboptimal solutions in sparse data conditions. Additionally, user ratings facilitate the learning of item content representations that better meet recommendation needs. MD-CVAE also incorporates a symmetric inference strategy, connecting the latent item embeddings in the decoder to the UAE encoder's first-layer weights, enabling the recommendation of new items without retraining (Zhu, 2022).

Li et al. proposed a Distributed Variational Autoencoder (DistVAE) for sequential recommendations, aiming to address data privacy concerns while achieving effective model training. DistVAE employs a client-server architecture, coordinating thousands of clients for training without requiring the aggregation of their raw data. DistVAE combines pseudo-batching for global model updates and a Gaussian Mixture Model (GMM) to dynamically cluster clients into virtual groups to improve the stability of global model training. Clients within each virtual group sequentially train “local” models, sharing training experiences to improve recommendation outcomes (Li, 2023).

3.3 Convolutional Neural Network-based

Convolutional Neural Networks (CNNs) are a type of feedforward neural network characterized by deep structures and convolutional computations, gaining significant attention in recent years for their efficiency in recognition tasks. The design of CNNs is inspired by the animal visual system, which processes information hierarchically to extract image features. An input layer, convolutional layers, pooling layers, and fully linked layers make up a CNN's conventional architecture. Local feature extraction is performed by the convolutional layers, feature dimensionality reduction is handled by the pooling layers, and classification and regression are managed by the fully connected layers (Shiri, 2023).

In recommendation systems, CNNs can achieve personalized recommendations by learning features from user behavior data. Their exceptional scalability allows them to efficiently handle sparse data that is high dimensional and on a wide scale, which helps to solve data sparsity and cold-start circumstances. The automatic feature learning capability of CNNs allows them to excel in multimedia tasks such as fashion recommendations, music streaming, and video content recommendations. Their hierarchical learning mechanism enables CNNs to capture a rich spectrum of information, from low-level to high-level features, enhancing robustness against variations in input data and further enhancing strength of recommendations.

Furthermore, Wang et al. suggested an automatic CNN recommendation system to create a system tailored for image classification tasks. This system analyzes the training data of classification tasks, evaluates the complexity of the task, and recommends the optimal CNN model based on the complexity score. This approach eliminates the need for extensive model training typically required in traditional model selection processes, thus saving time. The system also introduces a "capability score" to measure the classification capability of CNN models, taking into account factors such as computational cost, model depth and width, and the vanishing gradient problem (Wang, 2017).

To enhance recommendation performance by leveraging user reviews, Zheng et al. proposed a model called Deep Collaborative Neural Network (DeepCoNN). DeepCoNN is made up of two simultaneous neural networks: one learns properties connected to items based on their reviews, while the other utilizes user reviews to focus on user behavior. A shared layer connects these two networks, allowing latent factors to interact with each other. This joint modeling approach improves the accuracy of prediction, especially for users and items with limited ratings, effectively addressing the sparsity problem (Zheng, 2017).

In recent years, a novel neural network model called CoCNN has been developed for collaborative filtering (CF) and implicit feedback recommendation. CoCNN combines co-occurrence patterns with Convolutional Neural Networks (CNNs). It employs

a multi-task neural network structure to bridge user-item pairs and item-item pairs through co-occurrence relationships, capturing more useful information. Additionally, CoCNN operates directly on the embedding layer using a CNN architecture, rather than employing outer product operations, thereby addressing the data and space complexity issues associated with outer products (Chen, 2022).

To deal with data sparsity in recommendation systems and enhance reliability, Li et al. proposed a model called Auxiliary Review-based Personalized Attention Convolutional Neural Network (ARPCNN). ARPCNN employs a parallel CNN structure to process item reviews and user reviews separately. By using customized word- and review-level attention processes, it gives keywords and significant reviews a larger attention weight. Additionally, ARPCNN introduces a user auxiliary network (Aux-Net), which leverages reviews from similar users in trust relationships as auxiliary information. This helps extract features more accurately for user modeling, thereby improving recommendation performance (Li, 2022).

3.4 Recurrent Neural Network-based

Recurrent Neural Networks (RNNs) are a particular kind of recursive neural network that accepts sequence data as input. In contrast to standard Feedforward Neural Networks (FNNs), all of the nodes (recurrent units) in an RNN are connected in a structure resembling a chain, and recursion happens in the direction that the sequence is evolving. RNNs are predicated on the idea that human cognition is dependent on memory and prior experiences, giving the network the ability to "remember" previous information. This makes RNNs particularly well-suited for handling and predicting time dependencies and temporal information in sequence data, where there is an inherent order and dependency between data points (Shiri, 2023).

In many recommendation scenarios, user behaviors are sequential or session-based, implying that a user's past interactions with certain goods (such music, movies, or products) have an impact on the items they may interact with later. This recurrent structure allows RNNs to maintain and pass data from earlier time steps to the present one. As RNNs retain a hidden state that evolves as new inputs (user interactions) are processed, they can capture the temporal dependencies between a user's past behaviors and future preferences. This makes RNNs an ideal choice for modeling the dynamic characteristics of user sessions in recommendation systems.

3.4.1 Gated Recurrent Unit-based

An RNN variant called the Gated Recurrent Unit (GRU) adds gating techniques to regulate information flow, allowing it to more effectively process sequential data. The two gates in GRU are the update gate and the reset gate. The **update gate** identified which information needs to be updated, while the **reset gate** decides which information should be forgotten. This gating mechanism enables GRUs to recognize distant dependencies in sequential data more effectively, which has led to significant success in recommendation systems (Shiri, 2023).

By controlling the information flow, GRUs solve the issue of the vanishing gradient often encountered in traditional RNNs, permitting them to retain important information over longer sequences. This makes GRUs particularly well-suited for applications such as personalized recommendations, where understanding long-term user preferences or behavior patterns is crucial for generating accurate predictions (Yang, 2020).

In collaborative filtering tasks, Bansal et al. encoded text sequences into latent vectors, specifically using GRU trained end-to-end. In the scientific paper recommendation task, this approach significantly improved recommendation accuracy, particularly outperforming methods that ignore word order in cold-start situations. By leveraging the regularization effect of multi-task learning (combining metadata prediction and content recommendation), the network shared the text encoder between the recommendation and metadata prediction tasks, preventing overfitting in deep models. This approach further enhanced performance and effectively alleviated the sparsity issue in the rating matrix (Bansal, 2016).

To tackle the sequential recommendation challenge, Donkers et al. suggested to modeling the temporal dynamics of consumption sequences using GRU. Additionally, they introduced a novel gated structure with an extra input layer to explicitly represent each user's personalized information. This led to the design of a user-level GRU specifically for generating personalized next-item recommendations (Donkers, 2017). With this method, the model can represent the preferences of each unique user more accurately and more successfully customize recommendations depending on user activity patterns.

To enhance personalization in recommendation systems, Zeng et al. proposed an algorithm that utilizes a GRU network as the primary model to reduce the effects of multi-layer networks' overfitting. Additionally, they introduced an attention mechanism, allowing the recommendation model to more precisely extract key information from user data while minimizing interference from irrelevant data. The model also employs a variable-length mini-batch allocation technique to guarantee more comprehensive and dependable training data, which improves the accuracy of personalized recommendations (Zeng, 2022).

3.4.2 Long Short-Term Memory-based

Another specialized version of RNNs, Long Short-Term Memory (LSTM), was designed to solve the disappearing and expanding gradient issues that arise when processing lengthy data sequences. In recommendation systems, user behavior may span long sessions, necessitating the consideration of early interactions. User preferences may also evolve slowly, meaning that older interactions could still be relevant for recommendations. In such cases, LSTMs are particularly useful for modeling these dynamics (Yang, 2020).

The primary features of LSTM are its gating mechanisms and parameter sharing. The three gates in LSTM are the **forget gate**, **input gate**, and **output gate**, which allow the network to dynamically decide what information to retain or forget. These gates enable LSTMs to effectively handle long-term dependencies, allowing the network to remember information from much earlier inputs and use that information in the current output. Additionally, LSTM shares the same weights at each time step of the sequence, enabling the model to process sequences of arbitrary lengths (Smagulova, 2019).

Time-LSTM is a frequently used variant of LSTM designed for modeling user sequential behavior by introducing time gates to handle time intervals, which helps in better capturing both long-term and short-term user interests, thereby enhancing recommendation performance. By explicitly modeling the time intervals between user actions, Time-LSTM significantly improves the utilization of user behavioral information, leading to better recommendations (Zhu, 2017).

3.5 Generative Adversarial Network-based

Generative Adversarial Networks (GANs) achieve data generation and recognition through the competition and cooperation of two neural network models. The generator and the discriminator make up the fundamental structures.

- The **generator**'s job is to generate realistic data samples from random noise, typically Gaussian noise. A neural network generates samples that approximate data distribution from a random vector, aiming to confuse the discriminator into perceiving the generated data is authentic.
- The **discriminator** is another neural network in charge of differentiating between the generator's fictitious data and actual data. The input is a data sample, and it generates a scalar result indicating the probability of the input being real. The aim of the discriminator is to increase its accuracy in distinguishing real data from generated data (Gui, 2021).

3.5.1 GANs for Personalized Recommendations

Generative Adversarial Networks (GANs) are often used to construct representations of user interest and preference features in personalized recommendation systems. By training the generator model, it can produce feature vectors that align with users' interests, while the discriminator model helps distinguish the user's true interest features. The system can precisely record user preferences due to this approach, which produces more personalized recommendationsl.

The core of a personalized recommendation system lies in precisely understanding user interests. GANs can leverage historical user behavior data to generate a user interest profile, which includes features such as the user's areas of interest, preference types, and behavioral habits. The capacity of the recommendation engine to match content with the user's tastes can be greatly improved by these profiles (Gao, 2021).

Moreover, personalized recommendation systems often require large datasets, but real-world data may sometimes be insufficient. GANs are able to produce artificial data that replicates the distribution of real data, effectively performing data augmentation. By incorporating superior synthetic data into the current dataset, this method enhances the recommendation system's accuracy and performance (Wu, 2019).

3.5.2 GANs for Cold-Start Problems

For new users and products with insufficient interaction data (e.g., ratings or clicks), GANs can generate synthetic preferences based on partial information, such as demographic data or initial behaviors. These synthetic preferences simulate how users might interact with different items, allowing the system to provide appropriate recommendations immediately, thereby reducing the negative effects of data sparsity on fresh users and items.

In recommendation systems designed for new users, the generator produces synthetic user preferences for the cold-start users or items, while the discriminator assesses the reliability of these artificial preferences by comparing them with real user interaction data. The interaction between these two networks helps generate more realistic composite ratings for cold-start users, allowing the recommendation system to make better initial recommendations. By providing accurate suggestions even in the lack of adequate interaction data, this dynamic method greatly enhances the system's capacity to address cold-start issues (Chen, 2023).

4 CHALLENGES AND FUTURE DIRECTIONS

As data volumes continue to grow, more efficient and accurate recommendation algorithms are constantly required to meet business needs. However, the large-scale application of deep learning in recommendation systems still faces several challenges.

Firstly, deep learning models possess strong representational power, but this also means they require substantial amounts of training data to fully learn features and achieve high accuracy. For small-scale user bases or new products, insufficient data can lead to ineffective model training and unreliable recommendation outcomes. Due to their data dependence, deep learning models are computationally intensive, often requiring expensive GPU servers to support the necessary calculations. Without adequate hardware resources, training times can become prohibitively long, or the training may not complete at all, imposing significant financial demands on organizations.

Secondly, compared to traditional machine learning algorithms, deep learning is more complex and requires advanced skills in areas like deep learning architectures and optimization algorithms. At present, professionals with these skills are scarce, and recruiting and training costs are high. A key challenge for implementing deep learning projects is finding and retaining the right talent. Many organizations already have established big data infrastructures, and integrating deep learning requires seamless incorporation with these existing systems for tasks like data analysis and feature engineering. This often involves the development of additional tools or components to bridge the gap between the existing architecture and the deep learning framework, allowing for efficient integration that can support recommendation tasks.

Thirdly, the internal decision-making process of deep learning models is not transparent, making it difficult to explain how inputs affect outputs. This "black-box" nature can make it challenging to provide users with reasons for recommendations, potentially undermining trust and impacting user experience and satisfaction − particularly in scenarios where transparency is important. In addition, deep learning models involve numerous parameters and hyperparameters, which need to be continuously tuned throughout training to optimize performance. This process is not only time-consuming but also requires significant experience, as different hyperparameter choices can greatly affect outcomes. As such, tuning can be highly complex and resource-intensive, adding to the challenges of applying deep learning effectively.

While these obstacles together prevent deep learning from being widely used in recommendation systems, they also show that there is still a great deal of room for advancement and study in this area.

5 CONCLUSION

In conclusion, the application of deep learning technology has greatly enhanced recommendation systems. Although conventional techniques like collaborative filtering and content-based filtering have their advantages, they face difficulties with data sparsity, computational complexity, and the cold-start issue. Foundational deep learning models, including MLP, autoencoders, CNNs, RNNs, and GANs, offer powerful optimization capabilities that improve personalization, address cold-start scenarios, and enhance scalability. As deep learning continues to evolve, its combination with recommendation systems will result in even higher accuracy and efficiency, transforming the way users interact with digital platforms.

REFERENCES

Bansal, T., Belanger, D., & McCallum, A. (2016, September). Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 107-114).

Chen, C. C., Lai, P. L., & Chen, C. Y. (2023). ColdGAN: An effective cold-start recommendation system for new users based on generative adversarial networks. *Applied Intelligence, 53*(7), 8302-8317.

Chen, M., Ma, T., & Zhou, X. (2022). CoCNN: Co-occurrence CNN for recommendation. *Expert Systems with Applications, 195*, 116595.

Chen, M., Weinberger, K., Sha, F., & Bengio, Y. (2014, June). Marginalized denoising auto-encoders for nonlinear representations. In *International Conference on Machine Learning* (pp. 1476-1484). PMLR.

Chen, W., Cai, F., Chen, H., & Rijke, M. D. (2019). Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems (TOIS), 37*(4), 1-30.

Donkers, T., Loepp, B., & Ziegler, J. (2017, August). Sequential user-based recurrent neural network recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 152-160).

Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences, 10*(21), 7748.

Gao, M., Zhang, J., Yu, J., Li, J., Wen, J., & Xiong, Q. (2021). Recommender systems based on generative adversarial networks: A problem-driven perspective. *Information Sciences, 546*, 1166-1185.

Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering, 35*(4), 3313-3332.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 173-182).

Ko, H., Lee, S., Park, Y., & Choi, A. (2022). A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics, 11*(1), 141.

Khan, Z. A., Zubair, S., Imran, K., Ahmad, R., Butt, S. A., & Chaudhary, N. I. (2019). A new users rating-trend based collaborative denoising auto-encoder for top-N recommender systems. *IEEE Access, 7*, 141287-141310.

Kingma, D. P. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, L., Xiahou, J., Lin, F., & Su, S. (2023). DistVAE: Distributed variational autoencoder for sequential recommendation. *Knowledge-Based Systems, 264*, 110313.

Li, M., Zhang, Z., Zhao, X., Wang, W., Zhao, M., Wu, R., & Guo, R. (2023, April). AutoMLP: Automated MLP for sequential recommendations. In *Proceedings of the ACM Web Conference 2023* (pp. 1190-1198).

Li, Z., Chen, H., Ni, Z., Deng, X., Liu, B., & Liu, W. (2022). ARPCNN: Auxiliary review-based personalized attentional CNN for trustworthy recommendation. *IEEE Transactions on Industrial Informatics, 19*(1), 1018-1029.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE, 109*(3), 247-278.

Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., & Nikolenko, S. I. (2020, January). RecVAE: A new variational autoencoder for top-n recommendations with implicit feedback. In P*roceedings of the 13th International Conference on Web Search and Data Mining* (pp. 528-536).

Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473*.

Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics, 228*(10), 2313-2324.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 1096-1103).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research, 11*(12), 1103-1127.

Wang, H., Shi, X., & Yeung, D. Y. (2015, February). Relational stacked denoising autoencoder for tag recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29, No. 1).

Wang, S., Sun, L., Fan, W., Sun, J., Naoi, S., Shirahata, K., ... & Hashimoto, T. (2017, July). An automated CNN recommendation system for image classification tasks. In *2017 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 283-288). IEEE.

Wu, Q., Liu, Y., Miao, C., Zhao, B., Zhao, Y., & Guan, L. (2019, August). PD-GAN: Adversarial learning for personalized diversity-promoting recommendation. In *IJCAI* (Vol. 19, pp. 3870-3876).

Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016, February). Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (pp. 153-162).

Yang, S., Yu, X., & Zhou, Y. (2020, June). LSTM and GRU neural network performance comparison study: Taking Yelp review dataset as an example. In *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)* (pp. 98-101). IEEE.

Zeng, F., Tang, R., & Wang, Y. (2022). User personalized recommendation algorithm based on GRU network

model in social networks. *Mobile Information Systems, 2022*(1), 1487586.

Zhang, G., Liu, Y., & Jin, X. (2020). A survey of autoencoder-based recommender systems. *Frontiers of Computer Science, 14*, 430-450.

Zhao, C. M. K. M., & et al. (2017). Multivariate variational autoencoder for learning disentangled representations. In *Proceedings of the 34th International Conference on Machine Learning, 70*, 403-412.

Zheng, L., Noroozi, V., & Yu, P. S. (2017, February). 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* (pp. 425-434).

Zhou, K., Yu, H., Zhao, W. X., & Wen, J. R. (2022, April). Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022* (pp. 2388-2399).

Zhu, Y., & Chen, Z. (2022, April). Mutually-regularized dual collaborative variational auto-encoder for recommendation systems. In *Proceedings of The ACM Web Conference 2022* (pp. 2379-2387).

Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., & Cai, D. (2017, August). What to do next: Modeling user behaviors by time-LSTM. In *IJCAI* (Vol. 17, pp. 3602-3608).