Chapter 17
Recommendation Algorithm Based on Wide&Deep and FM

Songkun Zheng, Xian Li, Xueliang Chen, and Xu Li

Abstract Online learning is more and more popular because it is not limited by time and space. How to choose a suitable course from thousands of online courses is a great challenge faced by online learners, and online course recommendation came into being. The personalized recommendation algorithm analyzes the user's preferences by collecting some previous historical records of the user and other information, and generates recommendations for the user. Since Wide&Deep was proposed, due to its inherent ease of implementation, adaptability, and versatility, this approach has gained significant traction across various industry sectors. But its feature intersection method is not efficient. Sufficient feature engineering is required to provide informative features that can effectively distinguish objects. In this paper, the WD-FM model is proposed by combining Wide&Deep and factorization machine, and good results have been achieved through experimental demonstration.

17.1 Introduction

With the advent of the era of data explosion, it is necessary to correctly solve the previous problem of information scarcity and the current problem of information overload. In the face of massive data, it is very difficult to obtain the necessary information accurately and effectively. Therefore, information filtering techniques should be used. Currently, information filtering technology is mainly divided into search engine technology and recommendation system technology [1].

S. Zheng · X. Li · X. Chen
Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang, China

University of Chinese Academy of Sciences, Beijing, China

S. Zheng
e-mail: zhengsongkun20@mails.ucas.ac.cn

X. Li (✉)
Beijing Zhongke Zhihe Digital Technology Co., Ltd., Beijing, China
e-mail: lixu181@mails.ucas.ac.cn

R. Kountchev et al. (eds.), *Proceedings of International Conference on Artificial Intelligence and Communication Technologies (ICAICT 2023)*, Smart Innovation, Systems and Technologies 368, https://doi.org/10.1007/978-981-99-6641-7_17

Classification retrieval and search engines alleviate the problem of information overload. When the information classification is inaccurate or the user enters too few keywords, the user's retrieval time will be increased and the retrieval results will be affected. At present, many fields have begun to introduce personalized recommendation systems, and it is particularly important to convert large amounts of data into valuable information [2].

As an important tool for information filtering, personalized recommendations are a potential solution to the current information overload problem. The personalized recommendation algorithm analyzes the user's preferences by collecting some previous historical records of the user and other information, and generates recommendations for the user [3, 4]. The recommendation system can provide services that meet the individual needs of different users, improve the efficiency of users in finding knowledge from information, thereby effectively retaining users and making the website invincible.

The results of this paper can be summarized in the following two points:

1. We propose a model called WD-FM, which consists of Wide&Deep and FM (factorization machine).
2. We propose a new resource recommendation method based on WD-FM model. It has the function of feature memory, and separates low-order features and high-order features, and finally inputs them into the same output layer to improve the recommendation accuracy.

17.2 Related Work

Recently, researchers have done a lot of research in the field of recommendation, and the recommendation system model represented by collaborative filtering used in traditional recommendation systems [5]. Because it is difficult to solve problems such as data sparsity and cold start, the recommendation effect is unsatisfactory, especially when dealing with the huge amount of data generated at present.

Based on the assumption that user preferences are influenced by a small number of latent factors, matrix factorization is widely used in recommender systems and that an item's rating depends on how each of its characteristic factors is applied to user preferences.

MF (matrix factorization) decomposes the user item scoring matrix into the product of two or more low-dimensional matrices to achieve dimensionality reduction, and uses the low-dimensional spatial data mainly for non-negative matrix decomposition and matrix generalization decomposition Among them, the non-negative matrix factorization (NMF) method is to decompose the user's rating matrix $R_{n \times m}$ into two real-valued non-negative matrices $U_{n \times k}$ and $V_{k \times m}$, so that $R \approx U^{\mathrm{T}} V$.

Matrix decomposition is used for modeling. Usually, build a matrix of 1/0 values from the interaction data, and then decompose the matrix into two lower-dimensional matrices. One of the matrices has the same number of rows as the number of users, and each row represents a latent feature vector of a user. For example, in the literature [6], an $N \times D$ matrix C is used to represent the performance of users on the forum, where each row represents the user n who has posted at least once on the forum, and each column d represents the defined in the text A class label among the five behavioral dimensions of learners in the forum. Each entry C_{nd} of C is 1 if user n published at least one post assigned a content label of d, and 0 otherwise. Therefore, C is a matrix with a value of 1/0, and then the Bayesian non-negative matrix factorization (BNMF) method is used for C to generate user latent feature vectors. Literature [7] first regards the user's click, reading or use of resources as an interaction, thus forming a user-resource interaction matrix. The generalized matrix factorization (GMF) method is used to decompose it into hidden feature vectors of users and resources. In order to incorporate the characteristics of long-term interaction between users and resources, the model also combines long short-term memory (LSTM) to further generate fusion of users and resources. Hidden eigenvectors, and after combining the two eigenvectors, they share the same sigmoid output layer.

Under the background of big data, deep learning technology is more and more introduced into the core model design of the recommendation system, and the same is true for movie recommendation systems [8, 9]. Deep learning has brought a revolutionary impact to the recommendation system and can significantly improve the effect of the recommendation system. There are two main reasons. One is that deep learning greatly enhances the fitting ability of the recommendation model, and the other is that the deep learning model can utilize the model. The structure simulates different user behavior processes such as changes in user interest and user attention mechanisms. Deep crossing [10] was the first model proposed. Compared with multilayer perceptron (MLP), deep crossing adds an embedding layer between original features and MLP. Convert the input sparse features into dense embedding vectors, and then participate in the training in the MLP layer, which solves the problem that MLP is not good at dealing with sparse features. Convert the input sparse features into dense embedding vectors, and then participate in the training in the MLP layer, which solves the problem that MLP is not good at dealing with sparse features.

The Wide&Deep [11] recommendation model proposed by Google combines a deep MLP with a single-layer neural network, and at the same time gives the network good memory and generalization. Since its proposal, Wide&Deep has been widely used in the industry by virtue of its characteristics of easy to implement, easy to implement, and easy to transform.

17.3 Preliminary Knowledge

This section provides a brief review of how linear regression and multilayer perceptrons work.

17.3.1 Wide&Deep Model

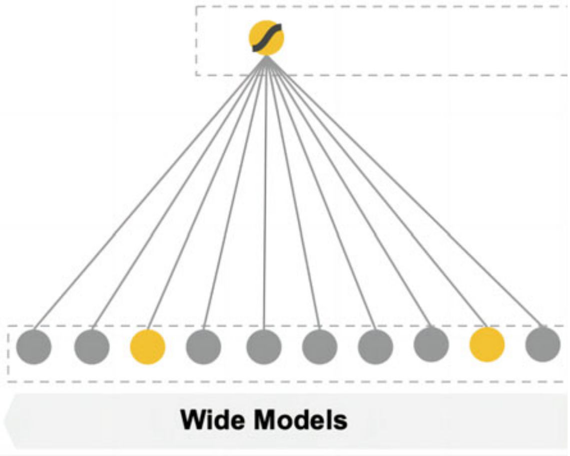
The Wide&Deep model is divided into the wide side and the deep side. The wide side mainly uses logistic regression in the generalized linear model. Generally speaking, it is the weight multiplied by the feature plus the bias, and then thrown into the sigmoid function, and finally the probability of predicting whether or not. Overall, the wide part of the Wide&Deep model creates the interaction between features through a linear model to provide the ability to model a wide range of features. By combining the nonlinear feature extraction capabilities of the deep part, the Wide&Deep model can simultaneously consider both wide and deep features in tasks such as recommender systems, thereby improving the model's expressiveness and prediction accuracy. Its structure diagram is shown in Fig. 17.1.

The logistic regression formula is as follows:

$$y = w^T x + b \tag{17.1}$$

y represents the final prediction result, x is a vector, which is a vector of n features, and w is the weight corresponding to each feature. b represents the bias. The feature set includes the original input features and transformed features.

Fig. 17.1 Wide side of the Wide&Deep model



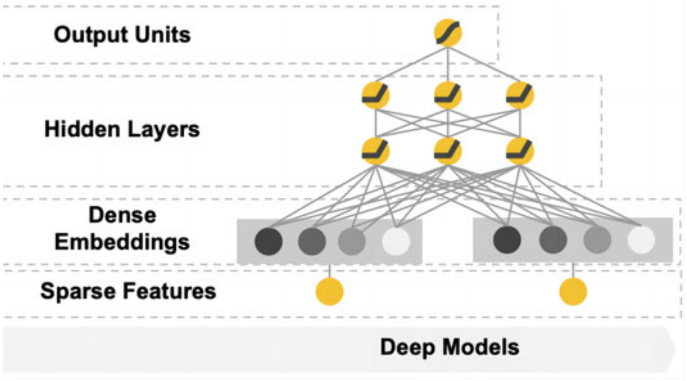


Fig. 17.2 Deep side of the Wide&Deep model

Among them, the optimizer of LR (logistic regression) is different from the past. In the past, stochastic gradient descent (SGD) was used, while the wide model used follow-the-regularized-leader (FTRL) published by Google on kdd in 2013. FTRL mainly fine-tunes the gradient. The idea is to hope that the new solution will not match the current solution. If the difference is too much, make the gradient step smaller. In addition, L1 regularization still needs to be added to make the solution found a little sparser.

The deep side is a multilayer perceptron, as shown in Fig. 17.2.

Each vector in the sparse and high-dimensional classification features is first converted into a low-dimensional and dense vector, which is called an embedding vector. The number of dimensions to join is typically on the order of O(10)–O(100). The embedding vectors are initialized randomly and trained concurrently during model training with the ultimate goal of minimizing the loss function. Low-dimensional dense embedding vectors are processed in the hidden layers of the feed-through neural network.

$$a^{(l+1)} = f\left(W^{(l)}a^{(l)} + b^{(l)}\right) \tag{17.2}$$

In the above formula, l represents the number of layers and f represents the activation function, usually a rectified linear unit (ReLU), where $a^{(l)}$, $b^{(l)}$, and $W^{(l)}$ are the activation function, bias, and model weights of layer l, respectively.

In the Wide part, after some nonlinear transformation and cross-combination of the input features, they are input into the linear model for modeling, and the interaction weights between the features are learned. The Deep part passes the input features through a series of hidden layers, and each hidden layer contains multiple neurons and performs feature transformation and abstraction layer by layer. Finally, after processing the depth part, a high-dimensional feature representation is obtained. The Wide&Deep model fuses the output of the breadth part and the depth part, which can be fused by simple weighted summation or other methods.

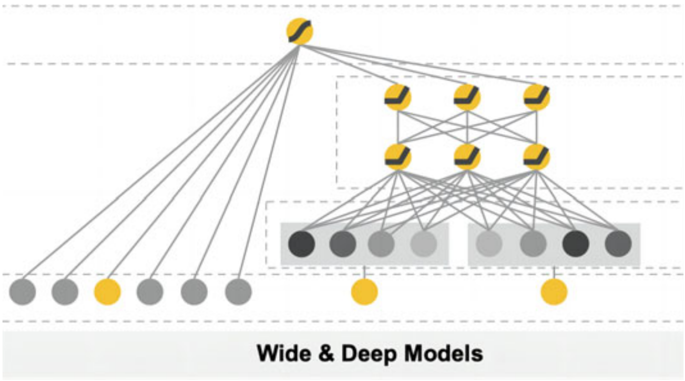


Fig. 17.3 Wide&Deep model

The combined model of wide and deep is shown in Fig. 17.3.

The prediction formula of the model is as follows:

$$P(Y = 1|x) = \sigma\left(w_{\text{wide}}^{T}[x, \emptyset(x)] + w_{\text{deep}}^{T} a^{(l_f)} + b\right) \tag{17.3}$$

In the above formula, Y is a binary class label, $\sigma(\cdot)$ represents the activation function, which is a sigmoid, $\varphi(x)$ represents the cross-product transformation of the initial feature x, and b represents a bias term. W_{wide} is the weight vector corresponding to the wide side vector, and W_{deep} is the weight to activate $a({}^{l}{}_{f})$ when computing.

17.3.2 FM Model

FM is a supervised learning method. It is mainly used for click-through-rate (CTR) estimation and is suitable for high-dimensional sparseness. The advantage is that it can automatically combine cross-features. The FM formula is as follows:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j \tag{17.4}$$

V_i is the hidden vector of the ith dimension feature, and < , > represents the vector dot product. $\hat{y}$ is the predicted output value for the sample. w_0 is the bias term, representing the global bias of the model. w_i is the linear weight of the ith feature, used to represent the contribution of the feature to the predicted output. x_i is the value of the ith feature in the sample x.

Feature combination is a problem encountered in many machine learning modeling processes. If you model directly, you may ignore the correlation information between features. Therefore, you can improve the effect of the model by building new cross-features. In fact, it is to increase the feature intersection term. In the general linear model, each feature is considered independently, without considering the relationship between features. But in reality, there are associations between a large number of features.

High-dimensional sparse matrix is a common problem in practical engineering, which directly leads to excessive calculation and slow update of feature weights.

The advantage of FM lies in the handling of these two aspects of the problem. The first is the combination of features, through the combination of two features, the introduction of cross-features to improve the model score. The second is the high-dimensional disaster, which estimates the characteristic parameters by introducing hidden vectors.

17.4 Recommendation Model

The problem of feature combination and feature intersection is very common. In practical applications, there are many more types of features, and the complexity of feature intersection is also much greater [12].

The key to solving this problem is the ability of the model to learn feature combinations and feature intersections. This is because it is the key to determining the model's ability to predict unknown features combined with the sample and measuring its recommendation effectiveness for complex recommendation problems.

Wide&Deep does not carry out special processing on feature intersection, but directly sends independent features into the neural network, allowing them to be freely combined in the network [13]. Such feature intersection methods are not efficient. Although neural network has strong fitting ability, the premise is that there are any multilayer network and any number of neurons.

In the case of limited training resources and limited time for parameter adjustment, MLP is actually relatively inefficient for feature intersection processing. MLP connects all features together into a feature vector through the concatenate layer, there is no feature intersection, and there is no relationship between two features [14].

FM is a classic traditional machine learning model for solving feature intersection problems. FM will use a unique layer FM layer to specifically deal with the intersection between features, and there are multiple inner product operation units in the FM layer to combine different feature vectors pairwise [15]. Through the inner product operation of the two features in the FM layer, the features can be fully combined. Combine Wide&Deep with FM to generate a new model with strong feature combination ability, fitting ability and memory ability.

The model as show in Fig. 17.4.

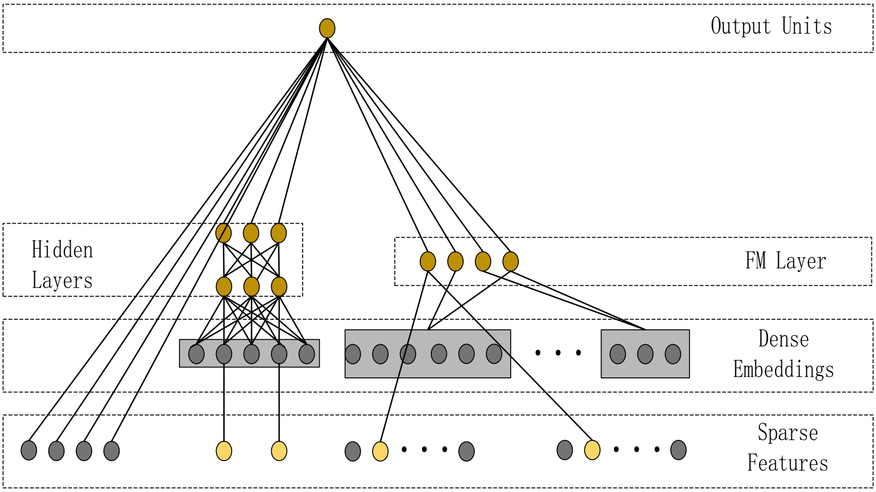


Fig. 17.4 WD-FM model

17.5 Experiments

17.5.1 Datasets

The dataset we used was the MovieLens dataset [16], on which the area under curve (AUC) and accuracy of our proposed Wide&Deep and FM model were evaluated. MovieLens: It is a non-commercial, research-oriented experimental site. The GroupLens research team created this dataset from data provided by users of the MovieLens website. This dataset consists of multiple movie rating datasets, each serving a different purpose. The collection is divided into several sub-datasets based on creation time, dataset size, etc. Each dataset varies in format, size, and purpose. This article uses the MovieLens 1 M dataset. MovieLens' 1 M dataset consists of 1,000,209 anonymous reviews of approximately 3900 movies. Users of these reviews joined MovieLens in 2000.

17.5.2 Evaluation Protocols

We divide the sample into training set and test set during evaluation, but splitting the sample is far from enough. In order to compare the quality of the model, we need to use some indicators to measure it.

We used the following evaluation metrics in the experiments in this paper: area under ROC (AUC) and logloss (cross-entropy).

The receiver operating characteristic (ROC) curve is a very commonly used indicator to measure the comprehensive performance of the model. ROC was first born in the military field, and then widely used in the medical field.

The x-axis in the ROC coordinate axis represents the false positive rate, and the y-axis in the curve represents the true positive rate.

The definitions of these two indicators are as follows:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} \tag{17.4}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} \tag{17.5}$$

In the above formula, P represents how many real positive samples there are, N represents the number of real negative samples, TP refers to how many P positive samples are predicted by the model as positive samples, and FP refers to how many N negative samples are classified as predicted by the model positive sample.

17.5.3 Baseline Algorithms

Wide&Deep: The Wide&Deep model includes two parts, namely the Wide part and the Deep part. The Wide side model is a generalized linear model, which can be expressed as $y = w^{\mathrm{T}} + b$. y represents the predicted output variable, w represents the weight vector, which is used to multiply the individual components of the input feature vector to weight them, and b stands for bias or intercept. The Deep side model is a typical deep neural networks (DNN) model.

DeepFM [17]: DeepFM is a combination of DNN and FM. On the basis of the Wide&Deep structure, FM is used to replace the LR of the Wide part, which can avoid artificially constructing complex feature projects. FM extracts low-level combined features, deep extracts high-level combined features, performs end-to-end joint training, and shares input embeddings.

17.5.4 Experimental Results

The main purpose of this experiment was to verify the accuracy of model by open resources data.

In this article, a high-performance server system with Ubuntu 20.04 LTS and a high-performance NVIDIA GeForce RTX 3080 graphics card was used. A deep learning framework is used called tensorflow-gpu-2.4.0. Use Pycharm under windows to connect to the server system remotely, and use Python language for software development. The specific experimental configuration is given in Table 17.1.

Table 17.1 Server software and hardware parameters

Name	Version
Operating system	Ubuntu 20.04 LTS
GPU	NVIDIA GeForce RTX 3080 GPU
CPU	Intel(R) Core(TM) i9-11900K CPU
Python	3.7.10
TensorFlow	2.4.0

Table 17.2 Comparison of models

Model	AUC	Logloss
Wide&Deep	0.7285	0.6065
DeepFM	0.6869	0.7721
WD-FM	0.7371	0.6010

As the amount of data increases, due to the addition of more redundant information, the accuracy of our prediction model begins to decline (Table 17.2).

In the mixed model structure of Wide&Deep, the wide side provides the model with a strong memory ability, and the deep side provides the model with a strong generalization ability. This structure allows the model to have both the advantages of logistic regression and the advantages of deep neural networks. In this way, a large number of historical behavioral characteristics can be memorized, and the ability to express has also been enhanced.

The prediction ability of the model for unknown feature combination samples depends on the feature combination and feature intersection. The wide part of DeepFM is FM, and FM can handle feature intersection well. There are multiple inner product operation units inside it. Two combinations, through this inner product operation, the features can be fully combined, and the prediction effect can be further improved. FM and Wide&Deep are combined to generate a brand new model with strong feature combination ability and strong fitting ability. Based on this, in order to make the model have memory ability, the three are combined. The memory ability can remember some rules in the data. Deep and FM can handle low-order feature combinations and high-order features well, and the prediction effect is compared with Wide&Deep and DeepFM. There are some improvements.

17.6 Conclusion

Recommendations are becoming more and more important in the era of big data. This paper proposes a new combinatorial recommendation model, called WD-FM. It combines the Wide & Deep and factorization machine (FM) models. Extensive experiments on the MovieLens dataset show that our model is improved in terms of effectiveness and accuracy.

References

1. Sedhain, S., Menon, A.K., Sanner, S., et al.: Autorec: autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on World Wide Web, pp. 111–112. United States (2015)
2. Mansur, F., Patel, V., Patel, M.: A review on recommender systems. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE, pp. 1–6. India (2017)
3. Ponnam, L.T., Punyasamudram, S.D., Nallagulla, S.N., et al.: Movie recommender system using item based collaborative filtering technique. In: International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), IEEE, pp. 1–5. India (2016)
4. Gupta, M., Thakkar, A., Gupta, V., et al.: Movie recommender system using collaborative filtering. In: International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, pp. 415–420. India (2020)
5. Alhijawi, B., Kilani, Y.: The recommender system: a survey. Int. J. Adv. Intell. Paradigms **15**(3), 229–251 (2020)
6. Gillani, N., Eynon, R., Osborne, M., et al.: Communication communities in MOOCs. arXiv preprint arXiv:1403.4640 (2014)
7. Li, J., Chang, C., Yang, Z., et al.: Probability matrix factorization algorithm for course recommendation system fusing the influence of nearest neighbor users based on cloud model. In: International Conference on Human Centered Computing, Springer, Cham, pp. 488–496. (2018)
8. Qu, Y., Cai, H., Ren, K., et al.: Product-based neural networks for user response prediction. In: IEEE 16th International Conference on Data Mining (ICDM), IEEE, pp. 1149–1154. Barcelona, Spain (2018)
9. Jais, I.K.M., Ismail, A.R., Nisa, S.Q.: Adam optimization algorithm for wide and deep neural network. Knowl. Eng. Data Sci. **2**(1), 41–46 (2016)
10. Shan, Y., Hoens, T.R., Jiao, J., et al.: 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, pp. 255–262 (2016)
11. Cheng, H.T., Koc, L., Harmsen J, et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. pp. 7–10 (2016)
12. Yuan, W., Wang, H., Hu, B., et al.: Wide and deep model of multi-source information-aware recommender system. IEEE Access **6**, 49385–49398 (2018)
13. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244 (2015)
14. Xu, J., Hu, Z., Zou, J.: Personalized product recommendation method for analyzing user behavior using DeepFM. J. Inf. Process. Syst. **17**(2), 369–384 (2021)
15. Chen J, Sun B, Li H, et al.: Deep CTR prediction in display advertising. In: Proceedings of the 24th ACM international Conference on Multimedia, pp. 811–820 (2016)
16. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. (TIIS) **5**(4), 1–19 (2015)
17. Guo, H., Tang, R., Ye, Y., et al.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017)