Deep Joint Learning Integrating Metadata and Attention Mechanism for Postprint Recommendation

Authors: Zhang Quangui, Li Zhiqiang, Zhang Xinxin, Cao Zhiqiang

Date: 2018-08-13T00:00:00+00:00

Abstract

Metadata-fused collaborative filtering recommendation, i.e., hybrid recommendation algorithms, are currently a hot research topic in the field of recommendation systems, capable of addressing data sparsity and cold start problems to some extent. However, most existing modeling methods for fusing metadata are built upon scenarios where user/item attributes have equal weights, resulting in non-prominent expression of key relationships between users and items, making it difficult to achieve satisfactory recommendation performance. To address these issues, we propose a deep joint learning recommendation method that fuses metadata and Attention mechanisms. It utilizes dual deep networks for joint learning: one network implements matrix nonlinear factorization based on implicit feedback data to learn personalized user/item relationships, while the other uses Attention mechanisms to automatically capture the impact of key user/item attributes on the recommendation task, modeling prominent user preference relationships by assigning different attribute weights and supplementing with an extended model. Experimental results demonstrate that the proposed recommendation algorithm exhibits relatively superior recommendation performance on two public datasets, MovieLens 100K and MovieLens 1M.

Full Text

Deep Joint Learning Recommendation based on Metadata and Attention Mechanism

Zhang Quangui, Li Zhiqiang, Zhang Xinxin, Cao Zhiqiang
School of Electronics & Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China

Abstract

Hybrid recommendation algorithms that integrate metadata into collaborative filtering have become a hot research topic in recommender systems, as they can alleviate data sparsity and cold-start problems to some extent. However, most existing metadata integration methods are built upon the assumption of equal weights for user/item attributes, resulting in insufficient expression of key relationships between users and items and limiting recommendation performance. To address these issues, this paper proposes a deep joint learning recommendation method that fuses metadata with an attention mechanism. The approach employs dual deep networks for joint learning: one network performs matrix nonlinear decomposition based on implicit feedback data to learn personalized user-item relationships, while the other utilizes an attention mechanism to automatically capture the influence of key user/item attributes on the recommendation task, extending the model by highlighting user preference relationships through assigning different attribute weights. Experimental results demonstrate that the proposed recommendation algorithm achieves superior performance on two public datasets, MovieLens 100K and MovieLens 1M.

Keywords: metadata; attribute weight; attention mechanism; deep joint learning; nonlinear factorization

0 Introduction

Matrix factorization models, also known as hybrid recommendation, can alleviate sparsity and cold-start problems to some extent. For instance, Fu et al. [?] measured potential personalized relationships between users and items from different perspectives in metadata and extended the matrix factorization model using these relationships for recommendation. Li et al. [?] proposed a similarity calculation method for item coupling relationships and integrated it into matrix factorization to improve recommendation performance. However, these existing methods are shallow models that cannot discover more abstract and complex interaction relationships. Recommender systems are technologies that alleviate information overload by estimating user needs and are widely applied in news websites, social media, and e-commerce. The concept of "recommender systems" was first introduced by Resnick in 1997, after which recommender systems attracted significant attention in both academia and industry. Various recommendation mechanisms have been proposed successively, becoming a hot research topic. Among existing methods, collaborative filtering recommendation has been extensively studied and extended, such as the traditional matrix factorization-based collaborative filtering algorithms proposed by Koren, Paterek, and Lee [?, ?, ?]. Researchers have since attempted to extend recommendation through metadata modeling. Nowadays, deep learning techniques for building deep models have been widely applied in recommender systems [?], as they can effectively capture nonlinear and complex user-item relationships to represent more sophisticated and abstract data structures. For example, Chang et al. [?] improved recommendation quality by applying neural network algorithms to traditional collaborative filtering. The CCCFNet in the literature [?] is also a dual-network approach (collaborative filtering and content information) that simulates user-item interactions through dot product at the top layer for recommendation. Guo et al. [?] proposed an end-to-end model that seamlessly integrates factorization machines and MLP, modeling high-order feature interactions through deep neural networks and low-order interactions via factorization machines. He et al. [?] also utilized multi-layer deep networks to replace matrix linear decomposition mechanisms, achieving nonlinear decomposition and addressing the limitations of linear MF to some extent. However, these models capture interactions among all attributes, whereas in practice, some interactions are unnecessary. As shown in Figure 1 [Figure 1: see original paper], when learning user/item group partitioning, user occupation and item category are key distinguishing features, while gender and region are non-essential components. Moreover, most existing metadata integration methods assume equal weights for user/item attributes throughout training, ignoring the varying influence of different attributes on recommendation and resulting in insufficient expression of key relationships between users and items, which significantly degrades recommendation performance. Consequently, researchers have gradually recognized that attributes play a decisive role regardless of how models are constructed.

In recent years, attention networks have been developed and gradually become a technical means for capturing global information. Consequently, neural networks based on attention mechanisms have become a hotspot across various domains and represent an effective solution to the aforementioned problems. For instance, Luong et al. [?] applied attention mechanisms in natural language processing to assign different weights to all words for translation tasks. The Google Mind team [?] used attention mechanisms in computer vision to automatically capture local image features for classification. Reference [?] proposed a fused deep neural network based on attention mechanisms, integrating user, author, and user interest similarity information between Twitter and user interests as learning attributes to predict user retweeting behavior. However, applications of attention mechanisms in recommendation remain scarce. We now apply this concept to recommender systems: assuming a user has five attribute values e_1, e_2, e_3, e_4, e_5 with recommendation weights w_1, w_2, w_3, w_4, w_5, without attention, the weights w_1 through w_5 are identical, yielding a total influence of $e_1 * w_1 + e_2 * w_2 + e_3 * w_3 + e_4 * w_4 + e_5 * w_5$. Conversely, with attention, the attention model learns different influence factors m_1, m_2, m_3, m_4, m_5 for the current user's attributes on the current recommendation, resulting in a total influence of $e_1 * m_1 * w_1 + e_2 * m_2 * w_2 + e_3 * m_3 * w_3 + e_4 * m_4 * w_4 + e_5 * m_5 * w_5$.

1 Algorithm Overview

The Deep Joint learning Recommendation based on Metadata and Attention mechanism (DJRMA) combines attention network feature learning and inference with deep network factorization machines to improve recommendation model performance by learning personalized relationships and mining implicit metadata relationships. As shown in Figure 2 [Figure 2: see original paper], the left decomposition mechanism network model is based on implicit feedback data and utilizes deep networks to replace matrix linear decomposition for nonlinear decomposition, thereby learning personalized preferences of users/items. The right model employs an attention mechanism to automatically capture the influence of key user/item attributes on the recommendation task, assisting in model extension by highlighting user preference relationship modeling through assigning different attribute weights. The core idea is to add a reconstructed user-item interaction matrix, formed by the product of denoised user personalized vectors and item personalized vectors, to the original user-item interaction matrix, thus creating a new joint interaction matrix for predictive recommendation.

Figure 1 [Figure 1: see original paper] illustrates the impact of different attribute proportions. In recent years, attention networks have been developed and gradually become a technical means for capturing global information. Consequently, neural networks based on attention mechanisms have become a hotspot across various domains and represent an effective solution to the aforementioned problems. For instance, Luong et al. [?] applied attention mechanisms in natural language processing to assign different weights to all words for translation tasks. The Google Mind team [?] used attention mechanisms in computer vision to automatically capture local image features for classification. Reference [?] proposed a fused deep neural network based on attention mechanisms, integrating user, author, and user interest similarity information between Twitter and user interests as learning attributes to predict user retweeting behavior. However, applications of attention mechanisms in recommendation remain scarce. We now apply this concept to recommender systems: assuming a user has five attribute values e_1, e_2, e_3, e_4, e_5 with recommendation weights w_1, w_2, w_3, w_4, w_5, without attention, the weights w_1 through w_5 are identical, yielding a total influence of $e_1 * w_1 + e_2 * w_2 + e_3 * w_3 + e_4 * w_4 + e_5 * w_5$. Conversely, with attention, the attention model learns different influence factors m_1, m_2, m_3, m_4, m_5 for the current user's attributes on the current recommendation, resulting in a total influence of $e_1 * m_1 * w_1 + e_2 * m_2 * w_2 + e_3 * m_3 * w_3 + e_4 * m_4 * w_4 + e_5 * m_5 * w_5$.

Figure 2 [Figure 2: see original paper] shows the experimental model of our recommendation algorithm.

1.1 Deep Decomposition Mechanism

The deep decomposition mechanism utilizes multi-layer deep networks to replace matrix linear decomposition for nonlinear decomposition based on implicit feedback data, thereby learning individual personalized relationships between users and items. The bottom input layer consists of two feature vectors describing user u and item i, which can be customized to support extensive modeling of users and items. In this paper, for clarity of explanation, we only adopt user/item IDs as universal input features, though the model can accommodate different inputs in practice. Above the input layer is the embedding layer, which, as shown in

Figure 3 [Figure 3: see original paper], maps sparse user/item ID vectors into dense vectors. The obtained user/item feature vectors can be regarded as latent features of users/items in the context of latent factor models.

Let $U \in \mathbb{R}^{M\times K}$ and $V \in \mathbb{R}^{N\times K}$ denote the latent factor matrices for users and items respectively, where M, N, and K represent the number of users, items, and latent features; θ_f denotes the model parameters of function f. Subsequently, sparse user (item) IDs are mapped through the embedding layer into dense vectors, i.e., user/item feature vectors. We use V_u and V_i as the feature vectors for user u and item i respectively. The forward process is defined as:

$$\psi_1(z) = \begin{bmatrix} V_u \\ V_i \end{bmatrix}$$

$$\psi_2(z) = \sigma(W_2' \psi_1(z) + b_2')$$

$$\psi_H(z) = \sigma(W_H' \psi_{H-1}(z) + b_H'), \text{ for } H \geq 2$$

where H is the depth of layers, σ is the activation function, and z_H, W_H', and b_H' are the output, model weights, and bias terms of layer H respectively. After learning through L layers, the final mapping function of the left model is:

$$\hat{y}_{ui} = \sigma(\psi_L(z))$$

1.2 Attention-Based Feature Learning and Inference Mechanism

The attention model is a brain-inspired model that simulates human attention, where at any given moment, the brain focuses on specific aspects of an object while ignoring others, representing a resource allocation model. It allocates more attention to important parts and less to others, rationally utilizing computational resources while eliminating the influence of non-critical factors. This paper introduces this concept, using the attention mechanism to automatically capture the influence of key user/item attributes on recommendation, highlighting user preference relationships by assigning different attribute weights, and then employing multi-layer deep network modeling to assist in extending the aforementioned nonlinear decomposition model, effectively preventing the loss of implicit metadata information.

As shown in Figure 4 [Figure 4: see original paper], the attention-based feature learning and inference mechanism represents user attribute features as one-hot encoded vectors and inputs them into the attention network to automatically capture key user/item attributes and infer their impact on recommendation. In our experiments, movie IDs are used as input, while user attributes utilize gender, age, and occupation information. Assuming we use the feature vector of user u, denoted as $A_u = \{A_{u1}, A_{u2}, ..., A_{ug}\}$, where g represents the number of user attributes and M is the number of users.

Figure 4 [Figure 4: see original paper] shows the attention-based local feature learning and inference mechanism. Therefore, we define the deep decomposition model as follows: $U \in \mathbb{R}^{M\times K}$ and $V \in \mathbb{R}^{N\times K}$ denote the latent factor matrices for users and items respectively, where M, N, and K represent the number of users, items, and latent features; θ_f denotes the model parameters of function f. Subsequently, sparse user (item) IDs are mapped through the embedding layer into dense vectors, i.e., user/item feature vectors. We use V_u and V_i as the feature vectors for user u and item i respectively. The forward process is defined as:

$$\psi_1(z) = \begin{bmatrix} V_u \\ V_i \end{bmatrix}$$

$$\psi_2(z) = \sigma(W_2' \psi_1(z) + b_2')$$

$$\psi_H(z) = \sigma(W_H' \psi_{H-1}(z) + b_H'), \text{ for } H \geq 2$$

where H is the depth of layers, σ is the activation function, and z_H, W_H', and b_H' are the output, model weights, and bias terms of layer H respectively. After learning through L layers, the final mapping function of the right model is:

$$\hat{y}_{ui} = \sigma(\psi_L(z))$$

The model framework's top-layer input comes from the outputs of both side models. We define the joint prediction function as:

$$\hat{y}_{ui} = \alpha \hat{y}_{ui}^{(1)} + (1-\alpha)\hat{y}_{ui}^{(2)}$$

where α is a parameter determining the training weight between the dual deep models, obtained through self-learning by the deep models. The algorithm formalizes the recommendation problem as a multi-classification problem for rating prediction, using the Softmax function as the final activation function and learning through multi-class logarithmic loss. With the above settings, we define the likelihood function as:

$$P(y, \hat{y}|U, V, \theta) = \prod_{(u,i)\in O} \hat{y}_{ui}^{y_{ui}} (1-\hat{y}_{ui})^{1-y_{ui}} \prod_{(u,i)\in O^-} \hat{y}_{ui}^{y_{ui}} (1-\hat{y}_{ui})^{1-y_{ui}}$$

where O denotes the set of observed interactions (as positive instances) and O^- denotes a uniformly sampled set of unobserved interactions (as negative instances). Taking the negative log-likelihood as the loss function yields:

$$L = -\sum_{(u,i)\in O\cup O^-} [y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})]$$

2 Experiments

2.1 Datasets

Currently, numerous public datasets are available for recommendation research. To validate the performance of our proposed deep joint learning recommendation algorithm fusing metadata and attention mechanisms, we employ two public datasets: MovieLens 100K and MovieLens 1M [?]. The MovieLens datasets are provided by the GroupLens research group at the University of Minnesota's Department of Computer Science and Engineering. Each user has rated at least 20 movies, with data sparsity levels of 0.9369 and 0.9553 respectively, and both provide auxiliary information such as user occupation and movie genre. MovieLens 100K contains 943 users, 1,682 movies, 19 movie genres, and 100,000 five-star ratings. The 1M dataset contains 6,040 users, 3,952 movies, 18 movie genres, and 1 million five-star ratings. Additionally, both provide auxiliary information like user occupation and gender. Although implicit feedback is easier to obtain than explicit feedback, publicly available datasets based on implicit feedback with user metadata are scarce. Therefore, following commonly used methods in literature [?, ?], we convert explicit rating data into implicit data.

2.2 Evaluation Metrics

The proposed algorithm utilizes deep networks for multi-class rating prediction. Currently, various metrics are used in the recommendation field to measure the quality of rating prediction. To evaluate item recommendation performance, we adopt the leave-one-out evaluation method, a widely used top-K evaluation approach in literature [?, ?, ?]. In experiments, we randomly sample 100 items that a user has not interacted with based on timestamps and rank the test item among these 100 items. The ranking list performance is measured by Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG) [?]. Unless otherwise specified, we set the truncation value K to 10 for both metrics. HR intuitively measures whether the test item appears in the top-10 list, while NDCG measures ranking quality by assigning higher scores to top-ranked positions. HR and NDCG are defined as follows:

$$HR@K = \frac{\#\text{hits}@K}{|GT|}$$

$$NDCG@K = \frac{1}{|GT|} \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

where GT denotes the set of test items and the algorithm-generated ranking list, and rel_i is the relevance of the item at position i. During evaluation, we use simple binary representation: $rel_i = 1$ if the item is in the test set, otherwise 0.

2.3 Experimental Environment and Parameter Settings

To demonstrate the superiority of our proposed deep joint learning recommendation model fusing metadata and attention mechanisms, we implement the algorithm on a single node of a cluster (experimental environment shown in Table 1). We compare it against the following representative baseline models:

a) CBMF [**?**]: Content-boosted matrix factorization model that directly incorporates metadata information into the matrix factorization framework.

b) NeuMF [**?**]: Neural network-based collaborative filtering method for comparing the impact of auxiliary information on recommendation effectiveness.

c) ItemKNN [**?**]: Standard item-based collaborative filtering method.

Table 1 Experimental Environment

Environment	Configuration
OS	Linux
CPU	3.5GHz
RAM	96.00GB
Python	3.5
Deep Learning Libraries	TensorFlow 1.2.0, Keras 2.0.5

2.4 Experimental Results and Analysis

For parameter determination, we employ 5-fold cross-validation using item attributes (movie genres) and user attributes (gender, age, and occupation). We use positive instances along with four uniformly sampled negative instances per positive instance as input, evaluated via multi-class logarithmic loss. The model is trained from scratch with a learning rate of 0.001, optimized using Adam, which offers the advantage of bias correction and a bounded learning rate per iteration to stabilize parameters. The network follows a tower structure where each successive higher layer halves the size, with layer dimensions of 128→64→32. ReLU is selected as the intermediate activation function due to its support for sparse activation, making the model less prone to overfitting. First, we evaluate our model's recommendation performance, with final results shown in Table 2 .

Table 2 Experimental Comparison on MovieLens Datasets (K=10)

Dataset\Metrics	Model	HR	NDCG
MovieLens 100K	ItemKNN	-	-
	CBMF	-	-
	NeuMF	-	-
	DJRMA	-	-
MovieLens 1M	ItemKNN	-	-
	CBMF	-	-
	NeuMF	-	-
	DJRMA	-	-

Second, we vary the layer structure to evaluate performance, equivalent to changing matrix factorization dimensions. As shown in Figures 6 [Figure 6: see original paper] and 7 [Figure 7: see original paper], the DJRMA model further improves recommendation metrics HR and NDCG by enhancing key attribute influence factors. Under equivalent settings on the 100K dataset with factor size 32, HR improves by 1.9%, 8.3%, and 13.2% compared to NeuMF, CBMF, and ItemKNN respectively, while NDCG improves by 4.6%, 9.5%, and 22.5%. Similarly, analysis of Figures 8 [Figure 8: see original paper] and 9 [Figure 9: see original paper] reveals that on the 1M dataset, HR improves by 0.9%, 4.8%, and 13.9% compared to NeuMF, CBMF, and ItemKNN respectively, with NDCG improving by 1.1%, 8.9%, and 18.2%. Thus, the proposed algorithm demonstrates superior recommendation performance over baseline methods.

Furthermore, comparing Figure 6 with Figure 8 and Figure 7 with Figure 9 shows that DJRMA's recommendation metrics are superior on the 1M dataset compared to 100K, with HR and NDCG improvements of 3.9% and 6.4%. Therefore, as the training set increases, DJRMA exhibits strong learning capability, though the overall improvement rate decreases, indicating that issues such as sample imbalance affect the model.

Figure 10 [Figure 10: see original paper] shows training loss on the MovieLens 1M dataset across different iterations. The model utilizes implicit feedback for binary classification recommendation, and we obtain optimal model parameters through training log loss. The figure demonstrates that training loss gradually decreases with increasing iterations until stabilization. The most effective number of iterations is approximately 10-20. In subsequent MovieLens 100K experiments, we observe similar trends, where additional iterations may cause model saturation and cease to improve the training loss metric.

3 Conclusion

This paper combines novel attention network feature learning and inference with deep network factorization machines, proposing a deep joint learning recommendation method that fuses metadata and attention mechanisms. One component achieves matrix nonlinear decomposition to learn personalized user/item relationships, while the other utilizes an attention mechanism to automatically capture the influence of key user/item attributes on recommendation, extending the model by highlighting user preference relationship modeling through different attribute weights. The algorithm formalizes a modeling approach for collaborative filtering to solve recommendation problems, ultimately achieving rating prediction through dual deep network joint learning. Finally, we validate our algorithm on two public datasets, MovieLens 100K and 1M. Experimental results demonstrate that the proposed recommendation algorithm not only exhibits superior recommendation performance compared to baseline methods but also maintains comparable learning efficiency.

In future work, we will consider introducing richer metadata information to mine the impact factors of cross-attribute relationships on relationship expression, thereby improving algorithmic recommendation quality. Additionally, we will explore constructing more reasonable recommendation frameworks using residual networks, convolutional networks, or other deep models [**?**] to comprehensively utilize features at different abstraction levels, which remains a hot topic in current recommendation research.

References

[1] Bell R, Koren Y, Volinsky C. The bellkor 2008 solution to the Netflix prize [R]. Korbell Teams Report to Netflix, 2008: 709-714.

[2] Paterek A. Improving regularized singular value decomposition for collaborative filtering [C]// Proc of KDDCup & Workshop, 2007: 319-325.

[3] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401: 788-791.

[4] Fu Bin, Xu Guangdong, Cao Longbing, et al. Coupling multiple views of relations for recommendation [M]// Advances in Knowledge Discovery and Data Mining. [S. l.]: Springer International Publishing, 2015: 732-743.

[5] Li Fangfang. Non-IID recommender system [D]. Beijing: Beijing Institute of Technology. 2014.

[6] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering [C]// Proc of International Conference on Machine Learning. New York: ACM Press, 2007: 791-798.

[7] Zhang Feng, Chang Huiyou. A collaborative filtering algorithm embedded BP network to ameliorate sparsity issue [C]// Proc of International Conference on Machine Learning and Cybernetics. 2005: 1839-1844.

[8] Lian Jianxun, Zhang Fuzheng, Xie Xing, et al. CCCFNet: a content-boosted collaborative filtering neural network for cross domain Recommender Systems [C]// Proc of International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee, 2017: 817-818.

[9] Guo Huifeng, Tang Ruiming, Ye Yunming, et al. DeepFM: a factorization-machine based neural network for CTR prediction [C]// Proc of 26th International Joint Conference on Artificial Intelligence. 2017: 1725-1731.

[10] He Xiangnan, Liao Lizi, Zhang Hanwang, et al. Neural collaborative filtering [C]// Proc of International World Wide Web Conference Committee. 2017: 3-7.

[11] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation [C]// Empirical Methods on Natural Language Processing. 2015: 31-41.

[12] Mnih V, Heess N, Graves A. Recurrent models of visual attention [C]// Proc of the 27th International Conference on Machine Learning. 2014: 2204-2212.

[13] Zhang Qi, Gong Yeyun, Wu Jindou, et al. Retweet prediction with attention-based deep neural network [C]// Proc of the 25th ACM International on Conference on Information and Knowledge Management. New York: ACM Press, 2016: 75-84.

[14] Harper F M, Konstan J A. The MovieLens datasets: history and context [J]. ACM Trans on Interactive Intelligent Systems, 2016, 5(4): 19.

[15] Hu Yifan, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets [C]// Proc of the 8th IEEE International Conference on Data Mining. Washington DC: IEEE Computer Society, 2008: 263-272.

[16] Pan Rong, Zhou Yunhong, Cao Bin, et al. One-class collaborative filtering [C]// Proc of the 8th IEEE International Conference on Data Mining. Washington DC: IEEE Computer Society, 2008: 502-511.

[17] Bayer I, He Xiangnan, Kanagal B, et al. A generic coordinate descent framework for learning from implicit feedback [C]// Proc of International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017: 1341-1350.

[18] He Xiangnan, Zhang Hanwang, Kan M Y, et al. Fast matrix factorization for online recommendation with implicit feedback [C]// Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2016: 549-558.

[19] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback [C]// Proc of the 25th Conference on Uncertainty in Artificial Intelligence. 2012: 452-461.

[20] Wang Hao, Wang Naiyan, Yeung D Y. Collaborative deep learning for recommender systems [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2015: 1235-1244.

[21] Forbes P, Zhu Mu. Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2011: 261-264.

[22] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms [C]// Proc of International Conference on World Wide Web. New York: ACM Press, 2001: 285-295.

[23] 刘海龙, 李宝安, 吕学强, 等. 基于深度卷积神经网络的图像检索算法研究 [J]. 计算机应用研究, 2017, 34(12): 3816-3819. (Liu Hailong, Li Baoan, Lu Xueqiang, et al. Image retrieval algorithm based on deep convolutional neural network [J]. Research of Computer Applications, 2017, 34(12): 3816-3819.)

Note: Figure translations are in progress. See original paper for figures.