Neural Collaborative Filtering

Recommendation Strategy using Deep Neural Networks

Parth Patel - 04/19/2022
Reference Article - X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, "Neural Collaborative Filtering," International World Wide Web Conference Committee , 2019. DOI:http://dx.doi.org/10.1145/3038912.3052569

Introduction

- This paper strives to develop techniques based on neural networks to tackle the key problems in recommendation collaborative filtering — on the basis of implicit feedback. Although some recent work has employed deep learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items and acoustic features of musics. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items.

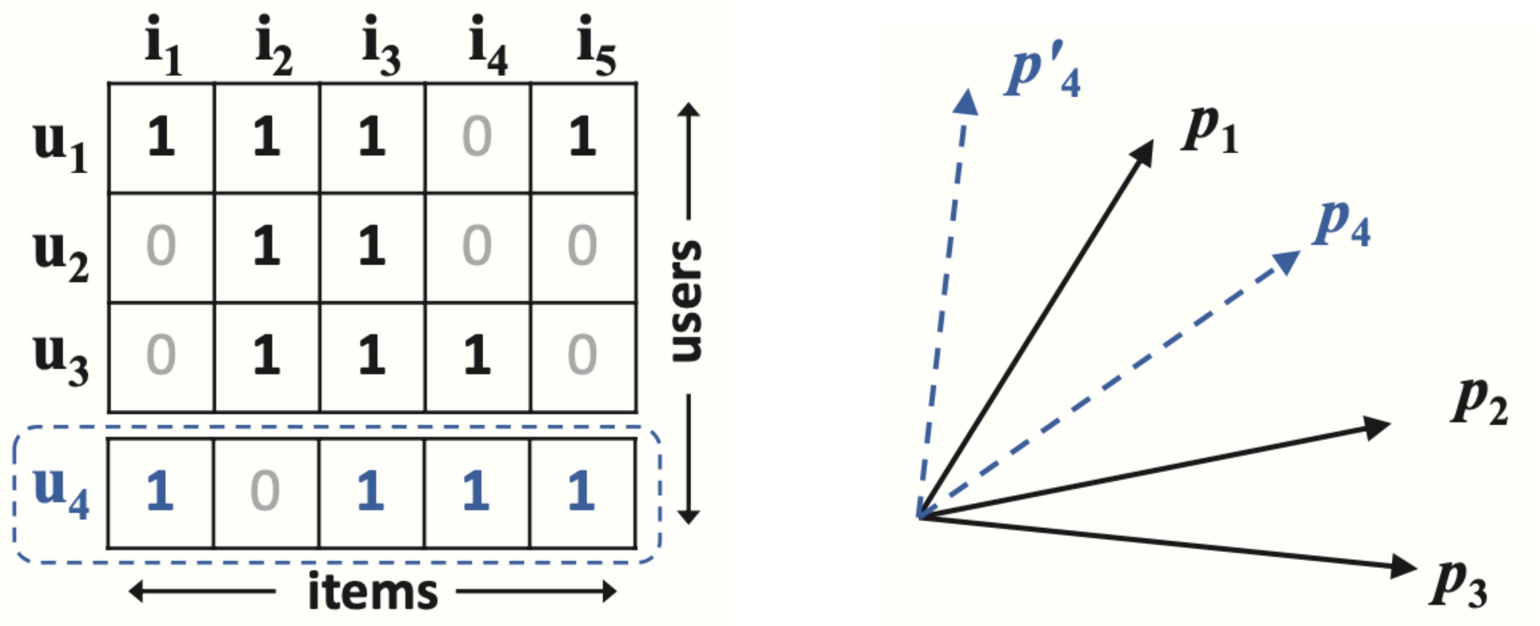
- By replacing the inner product with a neural architecture that can learn an arbitrary function from data, the paper present a general framework named NCF, short for Neural network- based Collaborative Filtering. NCF is generic and can express and generalize matrix factorization under its frame- work. To supercharge NCF modeling with non-linearities, the paper propose to leverage a multi-layer perceptron to learn the user–item interaction function

- The inner product, which simply combines the multiplication of latent features linearly, may not be sufficient to capture the complex structure of user interaction data. This paper explores the use of deep neural networks for learning the interaction function from data, rather than a handcraft that has been done by many previous work

Matrix Factorization

- Let M and N denote the number of users and items, respectively. We define the user–item interaction matrix Y ∈ R^{MxN} from users' implicit feedback as, $y_{ui} = 1$ - if interaction (user u, item i) is observed; $y_{ui} = 0$, otherwise.
- MF associates each user and item with a real-valued vector of latent features. Let p_u and q_j denote the latent vector for user u and item i, respectively, MF estimates an interaction y_{ui} as the inner product of p_u and q_j

$$\hat{y}_{ui} = f(u, i \,|\, p_u, q_j) = p_u^T q_i = \sum_{k=1}^{K} p_u k q_i k$$



(a) user–item matrix (b) user latent space

- An example illustrates MF's limitation. From data matrix (a), u_4 is most similar to u_1, followed by u_3, and lastly u_2. However in the latent space (b), placing p4 closest to p_1 makes p_4 closer to p_2 than p_3, incurring a large ranking loss

NCF General Framework

- To permit a full neural treatment of collaborative filtering, the paper adopt a multi-layer representation to model a user–item interaction you as shown in figure, where the output of one layer serves as the input of the next one. The bottom input layer consists of two feature vectors v_u and v_i that describe user u and item i, respectively

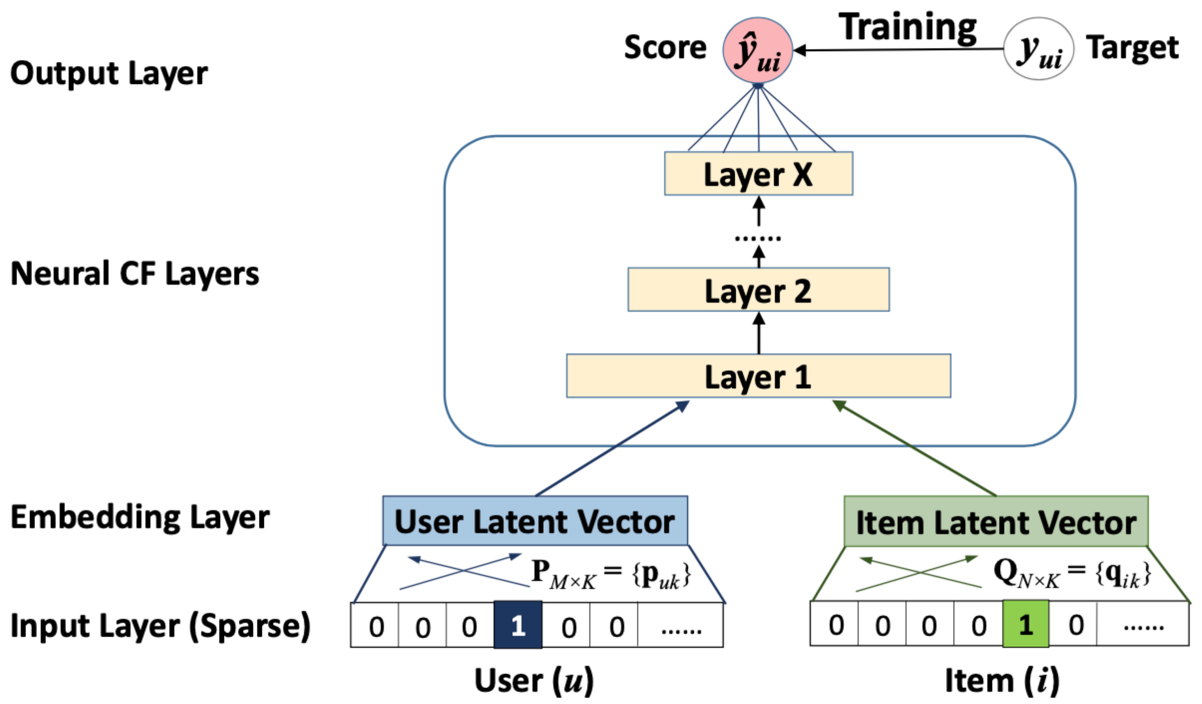


Figure 2: Neural collaborative filtering framework

- Above the input layer is the embedding layer, it is a fully connected layer that projects the sparse representation to a dense vector. The obtained user (item) embedding can be seen as the latent vector for user (item) in the context of latent factor model. The user embedding and item embedding are then fed into a multi-layer neural architecture, which we term as neural collaborative filtering layers, to map the latent vectors to prediction scores. Each layer of the neural CF layers can be customized to discover certain latent structures of user–item interactions. The dimension of the last hidden layer X determines the model's capability. The final output layer is the predicted score $y^\wedge_{ui}$, and training is performed by minimizing the point wise loss between $y^\wedge_{ui}$ and its target value y_{ui}

- The NCF's predictive model is formulated as

$$\hat{y}_{ui} = f(P^T v_u^U, Q^T v_i^I \mid P, Q, \theta_f)$$

Neural matrix factorization model

- The paper then extends their proposed method to combine MF with multi-layer perceptron as shown under

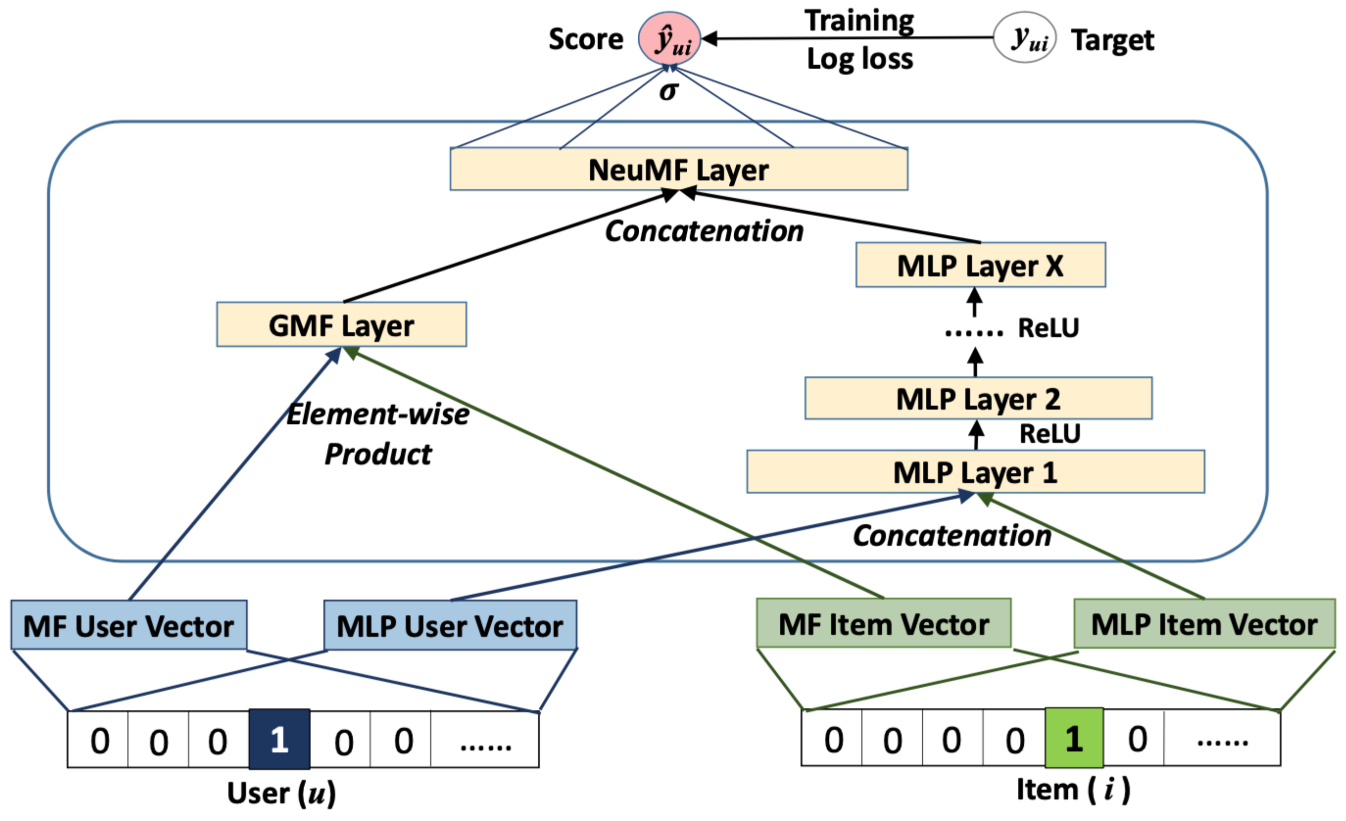


Figure 3: Neural matrix factorization model

- The paper allows MF and MLP to learn separate embeddings, and combine the two models by concatenating their last hidden layer, the formulation of which is given as follows

$$\hat{y}_{ui} = \sigma(h^T[\phi^{MF}\phi^{MLP}])$$

- Due to the non-convexity of the objective function of NeuMF, gradient-based optimization methods only find locally-optimal solutions. It is reported that the initialization plays an important role for the convergence and performance of deep learning models. Since NeuMF is an ensemble of GMF and MLP, the method propose to initialize NeuMF using the pre-trained models of GMF and MLP.
- First train GMF and MLP with random initializations until convergence, then use their model parameters as the initialization for the corresponding parts of NeuMF's parameters. The only tweak is on the output layer, where concatenate weights of the two models with a hyper parameter α determining the trade-off between the two pre-trained models