HO CHI MINH CITY UNIVERSITY OF SCIENCE
VIETNAM NATIONAL UNIVERSITY - HCMC

FINAL REPORT - RECOMMENDATION SYSTEM

Topic: Build an image recommendation system based on the Neuron Colaborative Filtering model

Lecturer: Mr Huynh Thanh Son
Group: UK

Group members

UK's members

Name	Student ID
Nguyen Thi Hoa	20280033
Nguyen Minh Phuc	20280074
Ngo Truong Minh Dat	20280013

Collaborative Filtering

- Recommendation systems, tailored to individual preferences, are widespread in various online services including E-commerce, advertising, and social networks. These systems focus on predicting a user's likelihood of choosing an item, relying on past interactions such as purchases and clicks.
- Collaborative Filtering (CF) tackles this by positing that users with similar behavioral patterns are likely to have comparable tastes in items.
- In this project, we employ Deep Neural Networks (DNNs) to learn the interaction function in the recommendation system to build an image recommendation system to recommend images for Pinterest users.

General framework

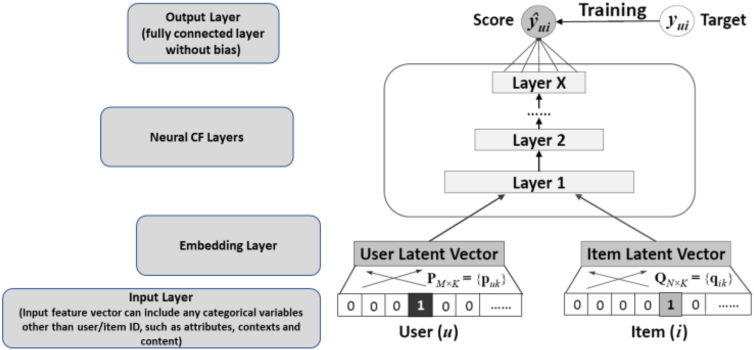

Figure 1: Neural collaborative filtering framework

General framework

We now formulate the NCF's predictive model as:

$$\hat{y}_{ui} = f(\mathbf{P}^T\mathbf{v}_u^U, \mathbf{Q}^T\mathbf{v}_i^I|\mathbf{P}, \mathbf{Q}, \theta_f)$$

where $\mathbf{P} \in \mathbb{R}^{M\times K}$ and $\mathbf{Q} \in \mathbb{R}^{N\times K}$, denoting the latent factor matrix for users and items, and θ_f denotes the model parameters of the interation function f.

Generalized Matrix Factorization (GMF)

Let define *Layer 1* as an element-wise product, and output layer as fully connected layer without bias, we have:

$$\hat{y}_{ui} = a_{out}(\mathbf{h}^T(\mathbf{p}_u \odot \mathbf{q}_i))$$

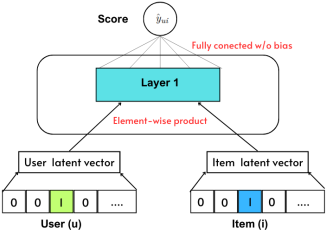

Figure 2: GMF architecture

where a_{out} and $\mathbf{h}$ denote the activation function (in this project we use sigmoid function) and edge weights of the output layer.

Multi-Layer Perceptron (MLP)

MLP can endow more non-linearities to learn the interaction function:

Layer 1:

$$\mathbf{z}_1 = \phi_1(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}$$

Remaining layers:

$$\phi_2(\mathbf{z}_1) = a_2(\mathbf{W}_2^T \mathbf{z}_1 + b_2)$$
$$\dots$$
$$\phi_L(\mathbf{z}_{L-1}) = a_L(\mathbf{W}_L^T \mathbf{z}_{L-1} + b_L)$$
$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \phi_L(\mathbf{z}_{L-1}))$$

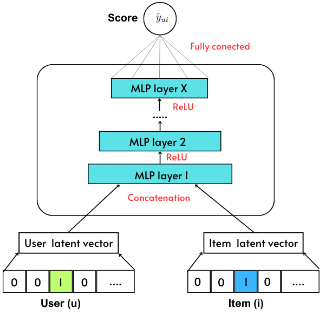

Figure 3: MLP architecture

where $\mathbf{W}_x$, $\mathbf{b}_x$ and a_x denotes weight matrix, bias vector and activation function for x-th layer's perceptron.

Fusion of GMF and MLP (NeuMF)

- GMF model: $\hat{y}_{ui} = \mathbf{h}^T a_{out}(\mathbf{p}_u \odot \mathbf{q}_i)$
- MLP model (1 linear layer): $\hat{y}_{ui} = \mathbf{h}^T a_{out}(\mathbf{W} \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix} + \mathbf{b})$
- The Neural Tensor Network naturally assumes GMF and MLP share the same embeddings, and combines theire latent space addition:

$$\hat{y}_{ui} = \mathbf{h}^T a_{out}(\mathbf{p}_u \odot \mathbf{q}_i + \mathbf{W} \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix} + \mathbf{b})$$

- **However**, sharing embeddings of GMF and MLP might limit the performance of the fused model. For example, it implies that GMF and MLP must use the same size of embeddings; for datasets where the optimal embedding size of the two models varies a lot, this solution may fail to obtain the optimal ensemble.

Fusion of GMF and MLP (NeuMF)

To provide more flexibility to be fused model, we allow GMF and MLP to learn separate embeddings, and combine two models by concatenating their last hidden layer.

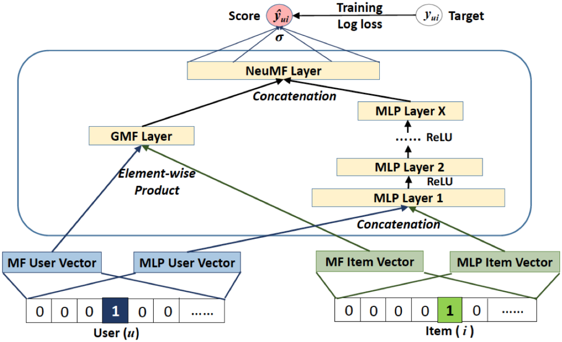

Figure 4: NeuMF architecture

Fusion of GMF and MLP (NeuMF)

The formulation of which is given as follows

$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G$$
$$\phi^{MLP} = a_L(\mathbf{W}_L^T(a_{L-1}(\dots a_2(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + \mathbf{b}_2) \dots)) + \mathbf{b}_L)$$
$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix})$$

where $\mathbf{p}_u^G$ and $\mathbf{p}_u^M$ denote the user embedding for GMF and MLP parts, and similar notation of $\mathbf{q}_i^G$ and $\mathbf{q}_i^M$.
This model combines the linearity of MF and non-linearity of DNNs for modelling user–item latent structures

Dataset

- We used Pinterest dataset - Learning Image and User Features for Recommendation in Social Networks for evaluating content-based image recommendation.
- This sub dataset also has files containing 60,000 users and over 1 million ratings, where each interaction denotes whether the user has pinned the image to her own board.

Evaluation Metrics

- **NDCG**: Normalized Discounted Cumulative Gain measure the effectiveness of search algorithms in ranking results by evaluating the quality of the ranking by considering the relevance and the order of the items in the list, with higher emphasis on the top-ranked items. **NDCG** is defined as:

$$NDCG_{@K} = \frac{1}{IDCG_{@K}} \sum_{i=1}^{K} \frac{2^{r_i - 1}}{log_2(i+1)}$$

Training Details

We used the batch size 256 and learning rate 0.001, number of epoch: 20.

- The number of MLP layer: 3.
- The number of predictive factors: [8,16,32,64], and the best number is 16
- Loss function: Mean Squared Error (MSE) is calculated by:.

$$\mathcal{L} = \sum_{(u,i)\in\mathcal{Y}\bigcup\mathcal{Y}^-} \left(y_{ui} - \hat{y}_{ui}\right)^2$$

Experiments Results

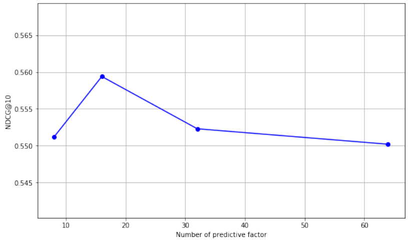

Figure 5: Performance of NDCG@10 $w.r.t.$ the number of predictive factors

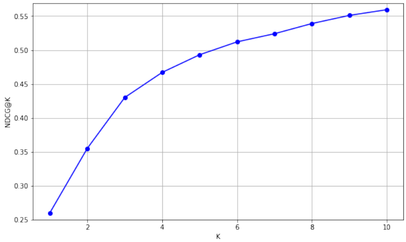

Figure 6: Evaluation of Top-K item reccomendation with 16 predictive factors

Conclusion

- In this project, we explored neural network architectures for collaborative filtering with instantiations —GMF, MLP and NeuMF —that model user–item interactions in different ways.
- The framework is simple and generic and achieve good performance.

Thanks for your listening!