## Two tower Network

Two tower networks are left and right towers encode {user,context } and {item} respectively. Twotower DNN is generalized from the multi-class classification neural network [19], a multi-layer perceptron (MLP) model, where the right tower of Figure 1 is simplified to a single layer with item embeddings.

Advantages :

1. label has structures or content features.
2. n contrast, with deep item tower, it is typically inefficient to sample and train on many negatives due to item content features and shared network parameters for computing all item embeddings
3. batch softmax optimization, where item probability is calculated over all items in a random batch, as a general recipe of training two-tower DNNs. if the data is streaming you would do a estimated item frequency
4. In recent years, deep neural networks (DNNs) have been shown effective in improving recommendation accuracy. Due to the nature of being highly nonlinear, DNNs offer a larger capacity for capturing complicated feature interactions
5. we leverage a two-tower neural network for modeling useritem interactions so that the inference can be conducted over a large corpus of items in sub-linear time; 2) learning NCF relies on point-wise loss (such as squared or log loss), while we introduce a multi-class softmax loss and explicitly model item frequency.

Modeling

* We consider a common setup for recommendation problems where we have a set of queries and items. Queries and items are representsed by feature vectors {xi } N i=1 and {yj } M j=1 respectively. Here xi ∈ X,yj ∈ Y are both mixtures of a wide variety of features (e.g., sparse IDs and dense features) and could be in a very high dimensional space. The goal is to retrieve a subset of items given a query. In personalization scenario, we assume user and context are fully captured in xi .
* The goal is to learn model parameter θ from a training dataset of T examples, denoted by T := {(xi ,yi ,ri)}T i=1 ,where (xi ,yi) denotes the pair of query xi and item yi , and ri ∈ R is the associated reward for each pair.
* Intuitively, the retrieval problem can be treated as a multi-class classification problem with continuous rewards. In classification tasks where each label is equally important, ri = 1 for all positive pairs. In recommenders, ri can be extended to capture various degrees of user engagement with a certain candidate. For example, in news recommendations, ri can be the time a user spent on a certain article

https://dl.acm.org/doi/pdf/10.1145/3366424.3386195

Case Study : IG explore

Goal - Explore to the interest of the user

Data - every interaction, based on prior interest, billions of options

Fundamental blocks - Query language to get data,

Representation choices, with changing catalog you can't do content based modeling, an account is represented by ig2vec, similar accounts using KNN, relevant account using cosine similarity

On ranking them for freshness and interest. Lightweight model is used to 500 ish piece to replicate a heavy weight model. Its objective function is to optimize for NDCG ranking (a measure of ranking quality) loss over main ranking model’s output

Main - candidate generation

→ look for similar content based on embeddings

→ filter out inappropriate content,

→ reduce it to 500

Ranking

→ distillation model light weight model

→ light weight wut h fill feature set

→ dense and sparse features

Modeling choice is MTML

Formula with scalars

General Reccomendation problem

Its two phase retrieval and ranking problem - That is, a scalable retrieval model first retrieves a small fraction of related items from a large corpus, and a fully-blown ranking model reranks the retrieved items based on one or multiple objectives such as clicks or user-ratings

General problem is → Given a triplet of {user,context,item}, a common solution to build a scalable retrieval model is: 1) learn query and item representations for {user,context } and {item} respectively; and 2) use a simple scoring function (e.g., dot product) between query and item representations to get recommendations tailored for the query

Traditionally, Matrix Factorization →