搜索推荐中的召回匹配模型综述(二)--基于表示学习的深度学习方法

辛俊波 浅梦的学习笔记 2020-01-10

“本文是搜索推荐中的召回匹配模型综述系列的第二篇，上一篇为搜索推荐中的召回匹配模型综述(一)--传统方法。
本文主要介绍了搜索推荐中基于representation learning的深度学习方法，包括基于协同过滤的方法(DMF,autoRec,协同降噪自编码器等)以及基于协同过滤+sideinfo的方法(DCF,DUIF,ACF,CKB)，并说明上述方法的结构范式和应用领域。”

作者：辛俊波
来源：知乎专栏 闲聊广告ctr预估模型。
编辑：happyGirl

Part0 基于representation learning的深度学习方法

终于要讲到激动人心的深度学习部分了。深度学习匹配模型从大致方向上可以分为两大类，分别是基于representation learning的模型以及match function learning的模型。

本章主要讲述第一种方法，representation learning，也就是基于表示学习的方法。这种方法会分别学习用户的representation以及item的representation，也就是user和item各自的embedding向量（或者也叫做隐向量），然后通过定义matching score的函数，一般是简单的向量点击、或者cosine距离来得到两者的匹配分数。整个representation learning的框架如图3.1所示，是个典型的user和item的双塔结构

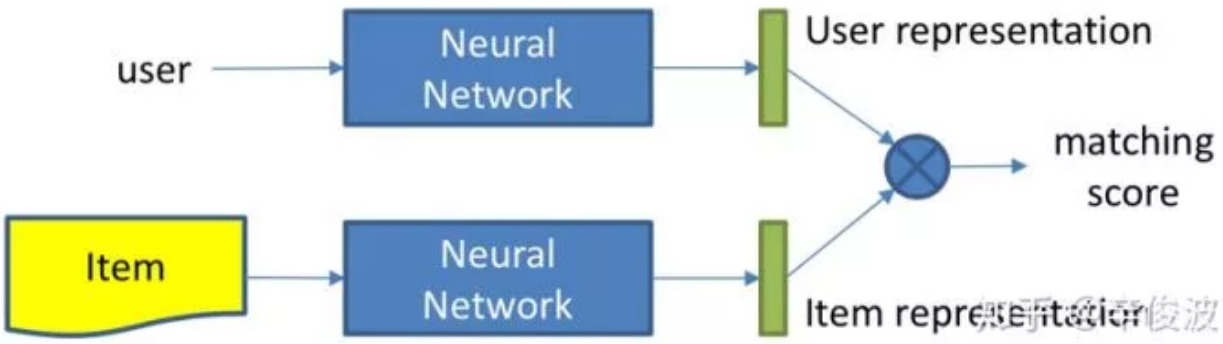


图3.1 基于representation learning的匹配模型

基于representation learning的深度学习方法，又可以分为两大类，基于CF以及CF + side info的方法。下面的介绍将分别从input 、representation function和matching function三个角度分别看不同的模型有什么不同

Part1 基于Collaborative Filtering的方法

CF模型（collaborative filtering）

重新回顾下传统方法里的协同过滤方法，如果从表示学习的角度来看，就是个经典的representation learning的模型，分别学习user和item的隐向量。

（1） Input layer

只有两个，分别是userid(one-hot)，itemid(one-hot)

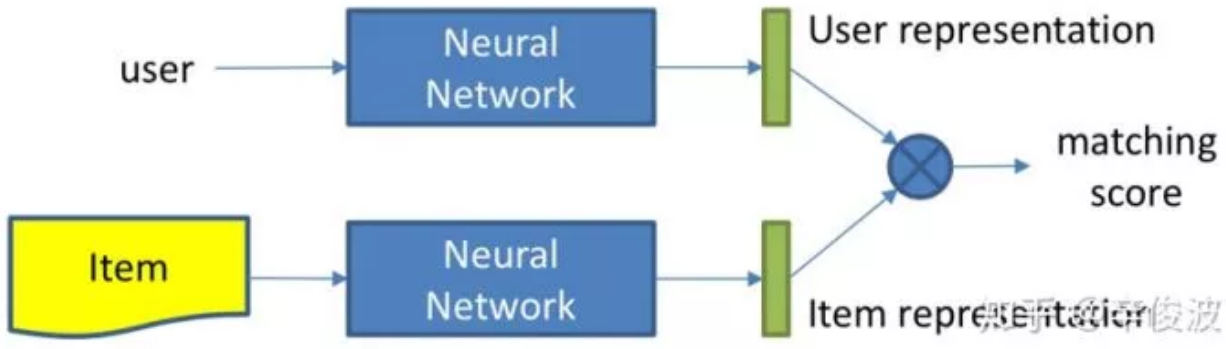
（2） representation function

线性embedding layer

（3） matching function

向量内积(inner product)

$$f_{MF}(u, i|\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u^\top \mathbf{q}_i = \sum_{k=1}^{K} p_{uk} q_{ik},$$



DMF模型（Deep Matrix Factorization）

DMF模型也就是深度矩阵分解模型，在传统的MF中增加了MLP网络，整个网络框架如图3.3所示。

（1）input layer

由两部分组组成，其中user由user交互过的item集合来表示，是个multi-hot的打分表示，如[0 0 4 0 0 ... 1 5 ...]，在矩阵中用行表示；item也由交互过的user集合来表示，也是个multi-hot的表示，如[5 0 0 3 ... 1 3]，在矩阵中用列表示

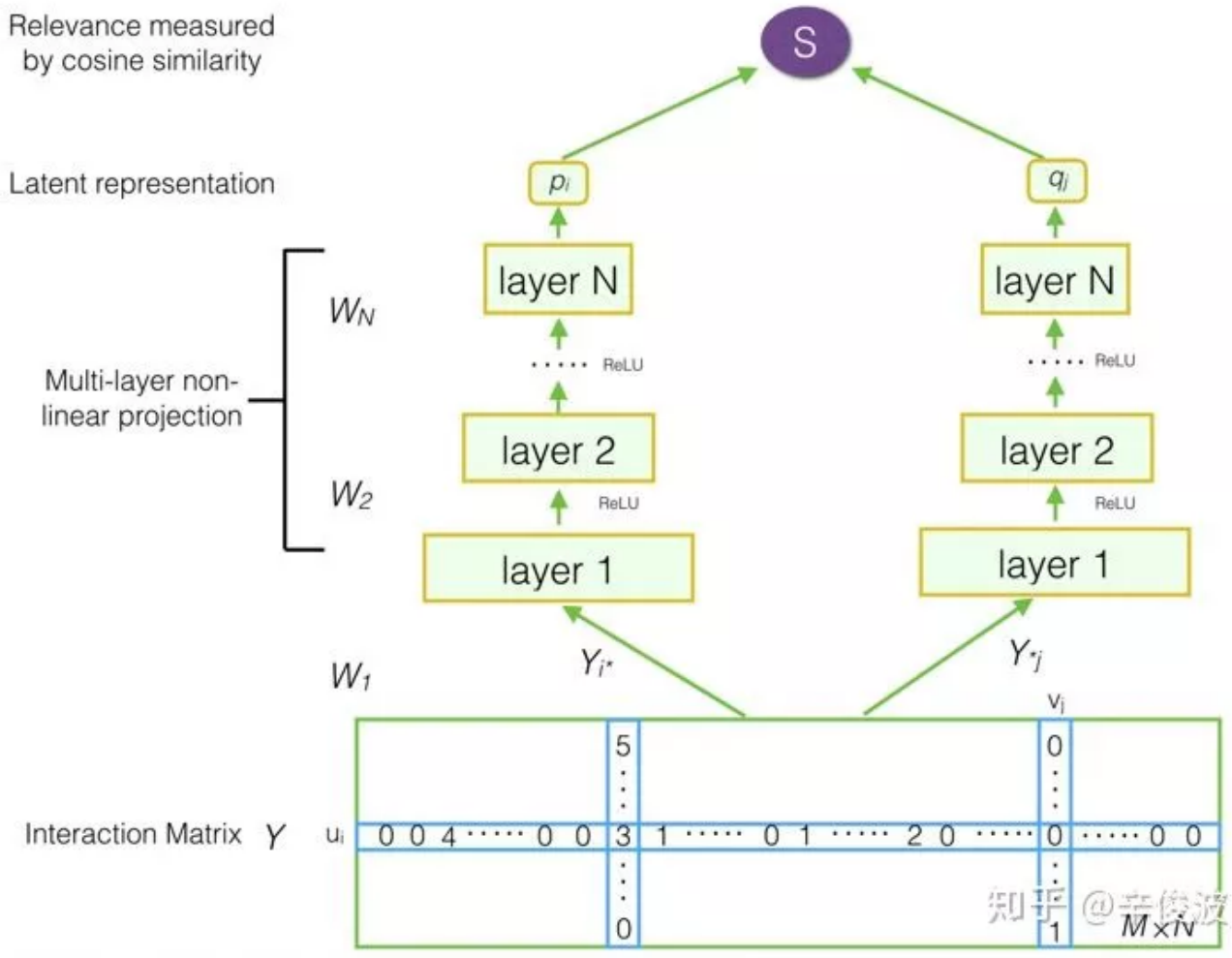


图3.3 DMF深度矩阵分解模型框架

可以发现这里的输入都是one-hot的，一般来说M用户数比较大，N作为item数量假设是百万级别的。

（2）representation function

Multi-Layer-Perceptron，也就是经典的全连接网络

（3）matching function

用cosine点击表示两个向量的匹配分数

$$\hat{Y}_{ij} = F^{DMF}(u_i, v_j|\Theta) = cosine(p_i, q_j) = \frac{p_i^T q_j}{\|p_i\| \, \|q_j\|}$$

对比普通的CF模型，最大的特点是在representation function中，增加了非线性的MLP，但是由于输入是one- hot的，假设用户规模是100万，MLP的第一层隐层是100，整个网络光user侧的第一层参数将达到1亿，参数空间将变得非常大

AutoRec模型

借鉴auto-encoder的思路，AutoRec模型对输入做重建，来建立user和item的representation，和CF一样，也可以分为user- based和item-based的模型。对于item-based AutoRec，input为R里的每列，即每个item用各个user对它的打分作为其向量描述；对于user-based AutoRec则是用R里的每行来表示，即每个user用他打分过的item的向量来表达。

用ru表示用户向量，ri表示item向量，通过autoencoder将ru或者ri投射到低维向量空间（encode过程），然后再将其投射到正常空间（decode过程），利用autoencoder中目标值和输入值相近的特性，从而重建（reconstruct）出用户对于未交互过的item的打分。

(1) input layer

和DMF一样，user用user作用过的item集合表示，item则用itemid本身表示，图中在原slides是说user- autoencoder，但个人在看原始autoRec论文时，这块应该有误，应该是item- based的，因为m表示的是用户数，n表示item数，下方的输入表示所有user(1,2,3,...m)对item i的交互输入

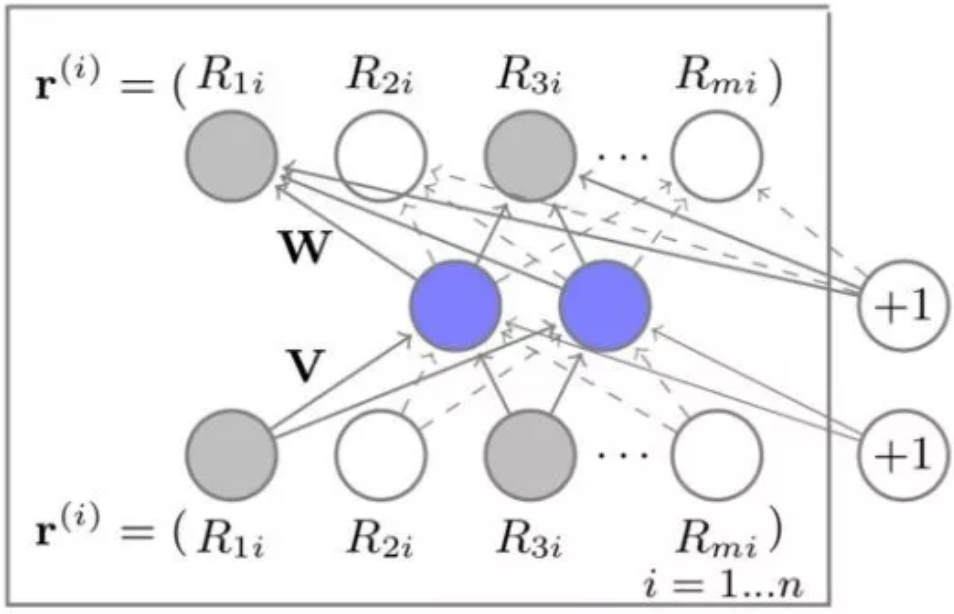


Figure 1: Item-based AutoRec model. We use plate notation to indicate that there are n copies of the neural network (one for each item), where $\mathbf{W}$ and $\mathbf{V}$ are tied across all copies.

图3.4 item-based的autoRec模型

（2）representation function

通过auto-encoder的结构表示，其中，h(r; theta)表示的是输入层到隐层的重建；由于输入的是用户交互过的item(multi- hot)，所以在隐层中的蓝色节点表示的就是user representation；而输出的节点表示的是item的representation，这样就可以得到user和item各自representation，如下面公式所示

$$h(\mathbf{r};\theta) = f\left(\mathbf{W}\cdot g(\mathbf{V}\mathbf{r}+\boldsymbol{\mu})+\mathbf{b}\right)$$

损失函数为最小化预测的平方差以及W和V矩阵的L2正则

$$\min_{\theta}\sum_{i=1}^{n}||\mathbf{r}^{(i)}-h(\mathbf{r}^{(i)};\theta))||_{\mathcal{O}}^{2}+\frac{\lambda}{2}\cdot(||\mathbf{W}||_{F}^{2}+||\mathbf{V}||_{F}^{2}),$$

（3）matching function

有了user和item的representation，就可以用向量点击得到两者的匹配分数

CDAE模型 (Collaborative Denoising Auto-Encoders)

CDAE模型类似SVD++的思想，除了userid本身表达用户，也将用户交互过的item作为user的表达。

（1） input layer

用户id,用户历史交互过的item；以及itemid。可以发现对比上述基础的autoRec，用户侧输入同时使用了用户历史交互过的item以及userid本身这个bias，思想很类似SVD++。如图3所示的input layer节点，绿色节点表示每个用户交互过的item，最下面的红色节点user node表示用户本身的偏好，可以认为是userid的表达

（2） representation function

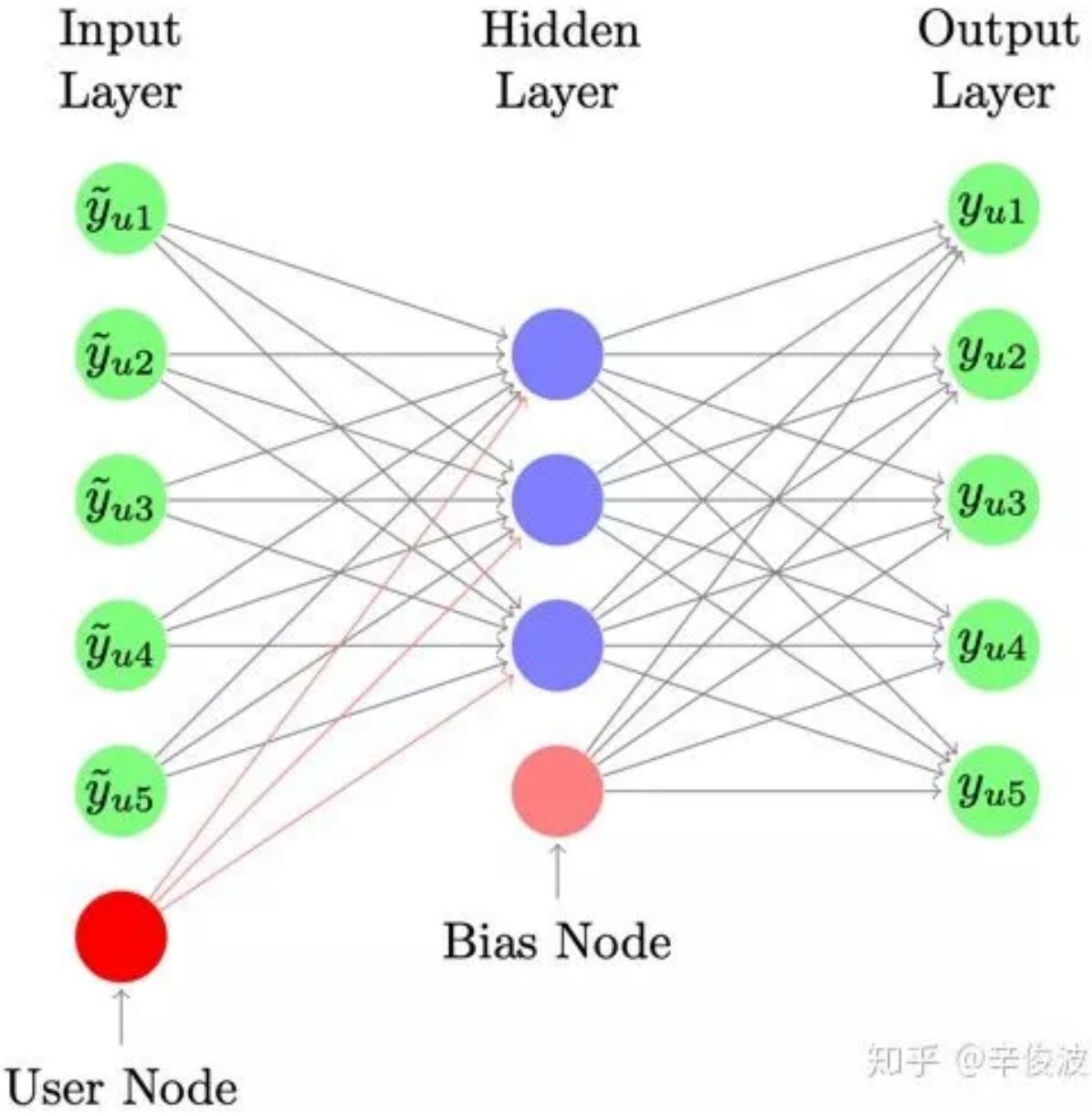


图3.5 CDAE模型结构

其中，中间蓝色的隐层节点作为用户表示，其中Vu为input layer中的user node的representation，针对所有用户id会学习一个和item无关的vu向量表达，可以认为是用户本身的bias，例如有些用户打分本身比较严格，再好的item打分也不会太高；有些用户打分很宽松，只要item别太差都会给高分，加上Vu可以更好的刻画用户之间天然的bias。

$$z_u = h\left(W^\top \tilde{y}_u + V_u + b\right),$$

而对于输出层的节点，可以认为是用户u对物品i的打分预测

$$\hat{y}_{ui} = f\left(\boldsymbol{W}_i'^{\top} \boldsymbol{z}_u + b_i'\right),$$

（3） matching function

使用向量点积作为匹配分数

$$\hat{y}_{ui} = \boldsymbol{W}'^{\top} \boldsymbol{V}_u,$$

基于CF方法的深度模型总结

总结下以上基于CF的方法，有以下几个特点

（1）user或者item要么由本身id表达，要么由其历史交互过的行为来表达

（2）用历史交互过的行为来作为user或者item的表达，比用id本身表达效果更好，但模型也变得更复杂

（3） Auto-encoder本质上等同于MLP+MF，MLP用全连接网络做user和item的representation表达

$$\underbrace{\text{MLP (representation learning)}}_{\text{Nonlinear}} + \underbrace{\text{MF (matching function)}}_{\text{Linear}}.$$

（4） 所有训练数据仅用到user-item的交互信息，完全没有引入user和item的side info信息

Part2 基于Collaborative Filtering+ side information的方法

基于CF的方法没有引入side information，因此，对于representation learning的第二种方法，是基于CF + side info，也就是在CF的方法上额外引入了side info。

DCF模型（Deep Collaborative Filtering）

(1) input layer

除了用户和物品的交互矩阵，还有用户特征X和物品特征Y

(2) representation function

和传统的CF表示学习不同，这里引入了用户侧特征X例如年龄、性别等；物品侧特征Y例如文本、标题、类目等；user和item侧的特征各自通过一个auto- encoder来学习，而交互信息R矩阵依然做矩阵分解U,V。整个模型框架如图3.6所示。

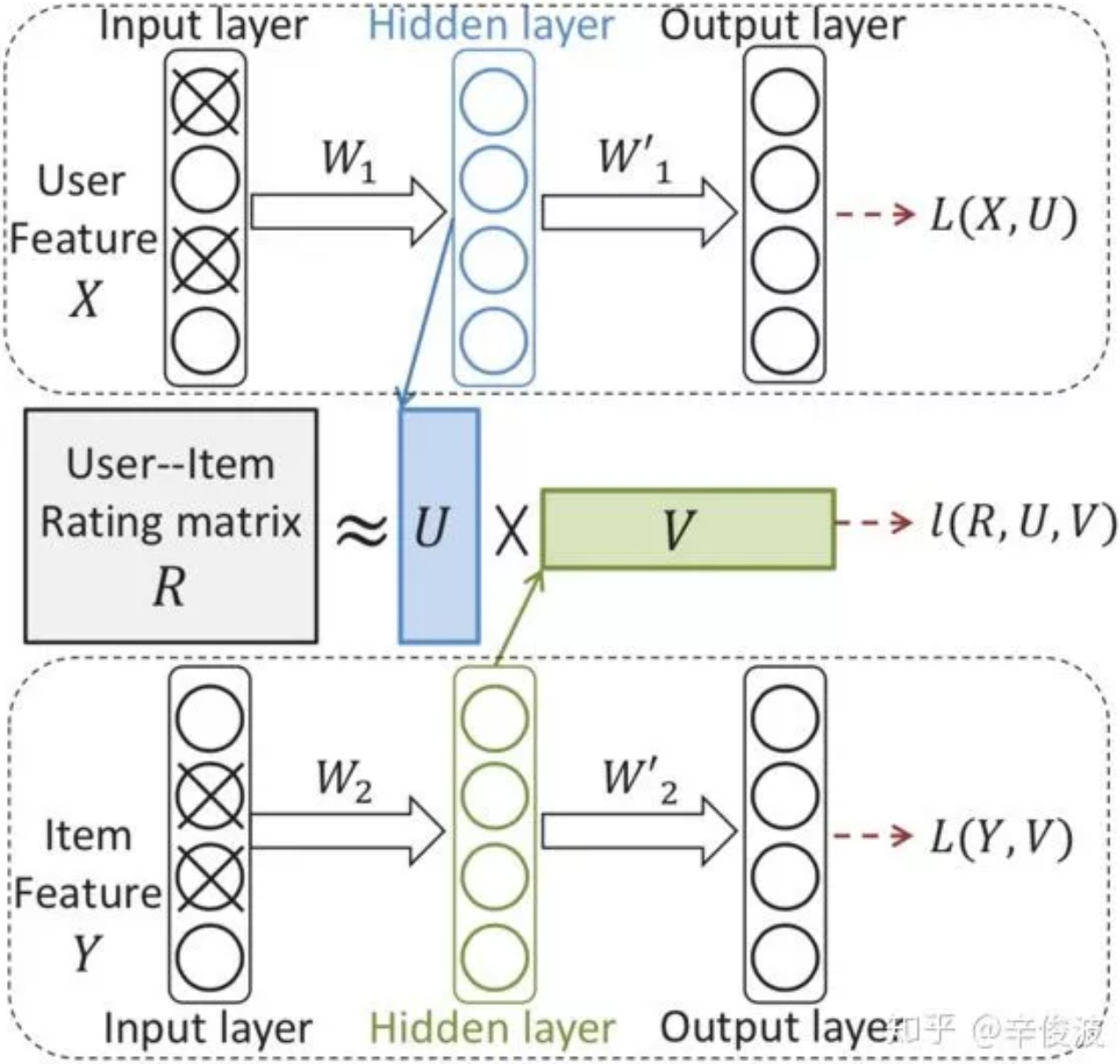


图3.6 DCF模型框架

其中W1, 表示的用户侧特征X在auto- encoder过程中的encode部分，也就是输入到隐层的重建，P1表示的是用户特征到交互矩阵R的映射；而W2表示物品侧特征Y在auto- encoder过程中的encode部分。P2表示的是物品特征到交互矩阵R的映射。

损失函数优化，需要最小化用户侧特征的reconstruction和item侧的encoder部分，以及交互矩阵和预测矩阵的平方差，还有加上L2正则。如图3.7第一个公式

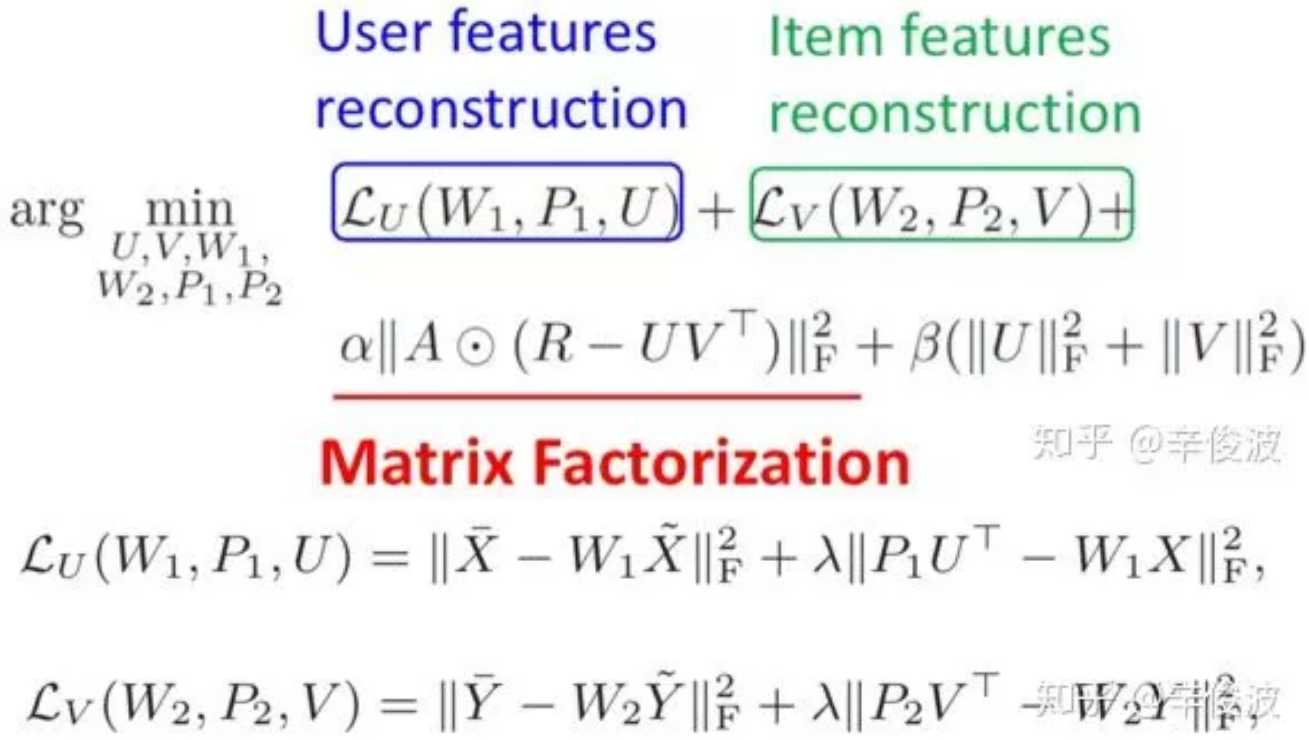


图3.7下面两组公式中，可以看出用户侧和物品侧特征都由两项error组成，第一项衡量的是input和corrupted input构建的预估误差，需要保证W1和W2对于corrupted 后的input x 和y不能拟合太差；第二项表达的是映射后的隐层特征空间W1X和投射到U矩阵的误差不能太大。

简单理解，整个模型的学习，既需要保证用户特征X和物品特征Y本身encode尽可能准确（auto- encoder的reconstruction误差），又需要保证用户对物品的预估和实际观测的尽可能接近（矩阵分解误差），同时正则化也约束了模型的复杂度不能太高

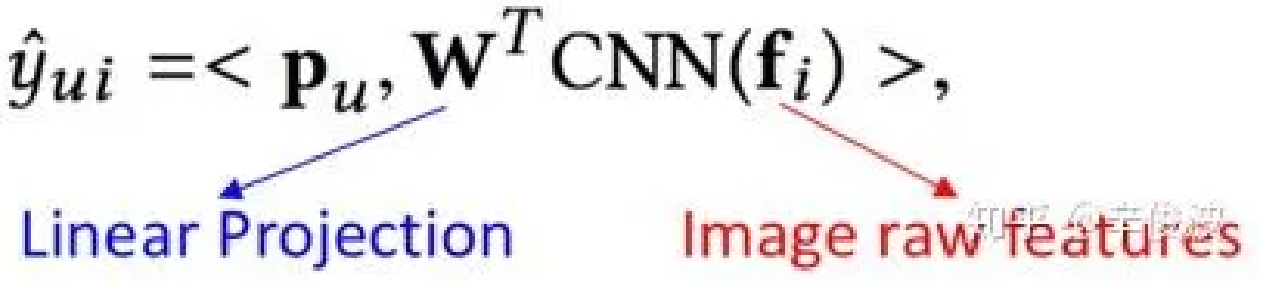
DUIF模型（Deep User and Image Feature Learning）

(1) input layer

除了用户和物品的交互矩阵，还有用户特征X和物品特征Y

(2) representation function

整个match score可以用下图表示：fi表示原始图片特征，通过CNN网络提取的图片特征作为item的表达，然后用一个线性映射可以得到item的embedding表达



(3) match function

通过模型学到的pu作为用户的representation，以及通过CNN提取的图片特征作为item的representation, 两者通过向量点积得到两者的匹配分数

ACF模型（Attentive Collaborative Filtering）

Sigir2017提出的Attention CF方法，在传统的CF里引入了attention机制。这里的attention有两层意思，第一层attention，认为用户历史交互过的item的权重是不一样的；另一个attention意思是，用户同一个item里到的视觉特征的权重也是不一样的，如图3.8所示。

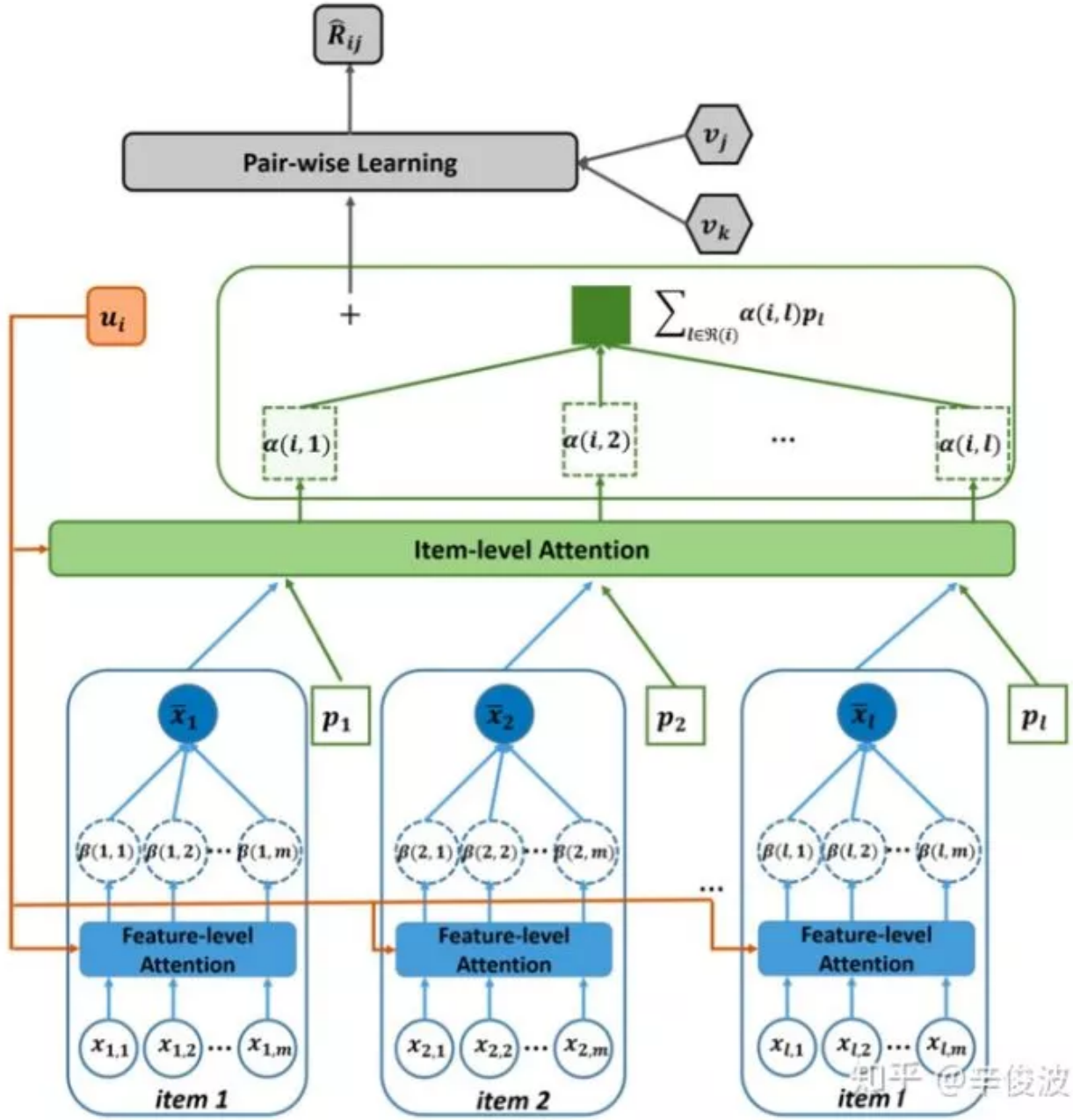


图3.8 ACF模型结构

(1) input layer

a) 用户侧：userid；用户历史交互过的item

b) Item侧：itemid; item相关的视觉相关特征

(2) representation function

可以分为两个attention，一个是component 层级的attention，主要是提取视觉特征；第二层是item层级的attention，主要提取用户对物品的喜好程度权重。

a) component-attention

在该paper里的推荐系统针对的是multi-media的，有很多图文和视频的特征信息提取，所以引入的第一层attention指的是component attention，认为对于不同的components 对item

representation的贡献程度是不同的，如图3.9所示

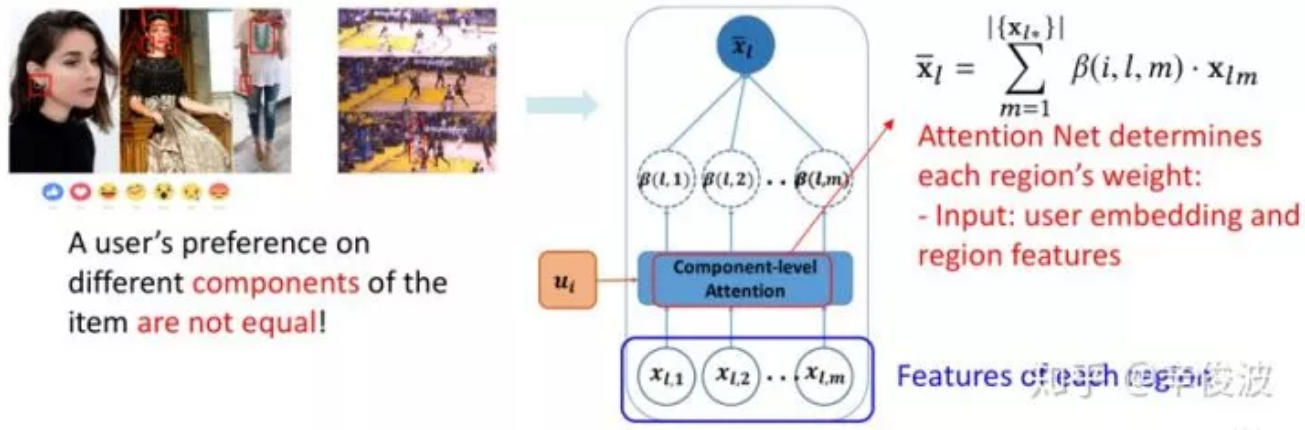


图3.9 component attention框架

对第l个item，输入为不同region本身的特征xl1, xl2, xlm，表示的是m个不同的item feature, 以及用户输入ui，最终item的表达为不同的region的加权embedding。

$$b(i,l,m) = \mathbf{w}_2^T \phi(\mathbf{W}_{2u}\mathbf{u}_i + \mathbf{W}_{2x}\mathbf{x}_{lm} + \mathbf{b}_2) + \mathbf{c}_2,$$

$$\beta(i,l,m) = \frac{exp\,(b(i,l,m))}{\sum_{n=1}^{|\{\mathbf{x}_{l*}\}|} exp\,(b(i,l,n))}.$$

其中第一个公式表示用户i对物品l第m个component（例如图片特征中的局部区域特征，或者视频中不同帧的特征）的权重；第二个公式softmax对attention权重归一化

b) item attention

第二层attention，认为用户作用过的item历史中，权重应该是不同的。这里文章使用了SVD++的方式，用户本身的表达引入了a(i, l)，代表的是用户i对其历史交互过的物品l的权重.

用户i对第l个item的权重表达可以用如下的数据表示：

$$a(i,l) = \mathbf{w}_1^T \phi(\mathbf{W}_{1u}\mathbf{u}_i + \mathbf{W}_{1v}\mathbf{v}_l + \mathbf{W}_{1p}\mathbf{p}_l + \mathbf{W}_{1x}\bar{\mathbf{x}}_l + \mathbf{b}_1) + \mathbf{c}_1$$

$$\alpha(i,l) = \frac{exp\,(a(i,l))}{\sum_{n\in\mathcal{R}(i)} exp\,(a(i,n))}.$$

其中ui是用户本身的latent vector, vl是物品l的latent vector，pl是物品l的辅助latent vector; xl是表示前面提到的从图文信息提取的特征latent vector。用户最终的表达是自身ui的latent vector，以及历史行为的attention加权的representation表示。

$$\hat{R}_{ij} = \left(\mathbf{u}_i + \sum_{l \in \mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l \right)^T \mathbf{v}_j$$

模型使用的是pairwise loss进行优化

$$\arg \min_{\mathbf{U},\mathbf{V},\mathbf{P},\Theta} \sum_{(i,j,k) \in \mathcal{R}_B} -\ln \sigma \left\{ \left(\mathbf{u}_i + \sum_{l \in \mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l \right)^T \mathbf{v}_j - \left(\mathbf{u}_i + \sum_{l \in \mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l \right)^T \mathbf{v}_k \right\} + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{P}||^2),$$

(3) representation function

使用user和item的向量点击作为匹配分数

CKB模型（Collaborative Knowledge Base Embedding）

CKB模型是在2016年KDD提出的，利用知识图谱做representation learning，模型框架如图3.10所示。整个CKB模型框架其实思想比较简单，分别在结构化信息、文本信息和视觉信息中提取item侧特征作为item的representation

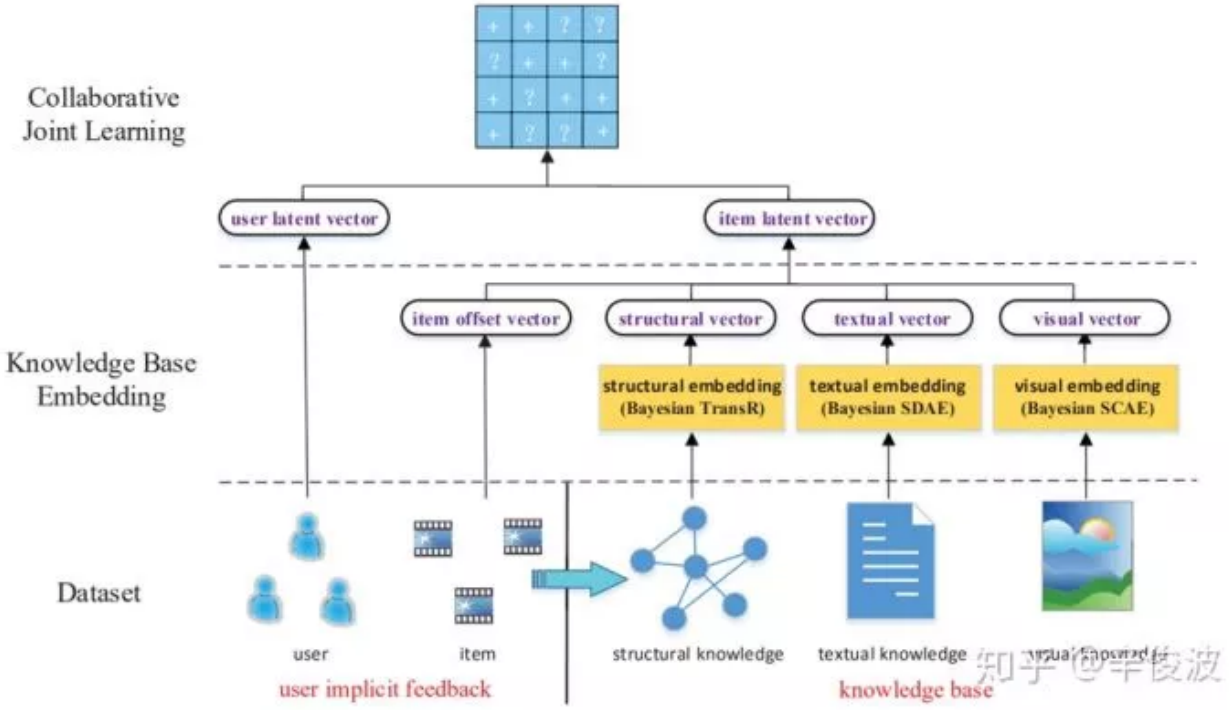


图3.10 CKB模型框架

(1) input layer

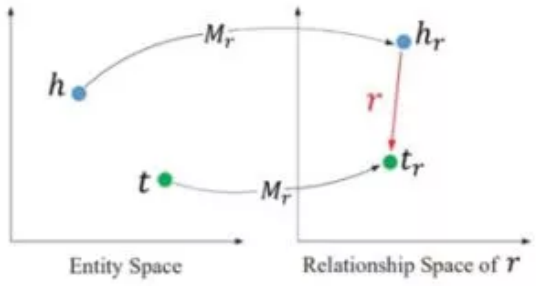
a) user侧：userid

b) item侧：itemid; 基于知识图谱的item特征（structural, textual, visual ）

(2) representation function

主要是从知识图谱的角度，从结构化信息，文本信息以及图文信息分别提取item侧的表达，最终作为item的embedding

a) 结构化特征struct embedding: transR, transE



$$\mathbf{v}_h^r = \mathbf{v}_h\mathbf{M}_r, \qquad \mathbf{v}_t^r = \mathbf{v}_t\mathbf{M}_r.$$

$$f_r(v_h, v_t) = ||\mathbf{v}_h^r + \mathbf{r} - \mathbf{v}_t^r||_2^2.$$

图3.11 struct embedding框架

b) 文本特征Textual embedding: stacked denoising auto-encoders (S-DAE)

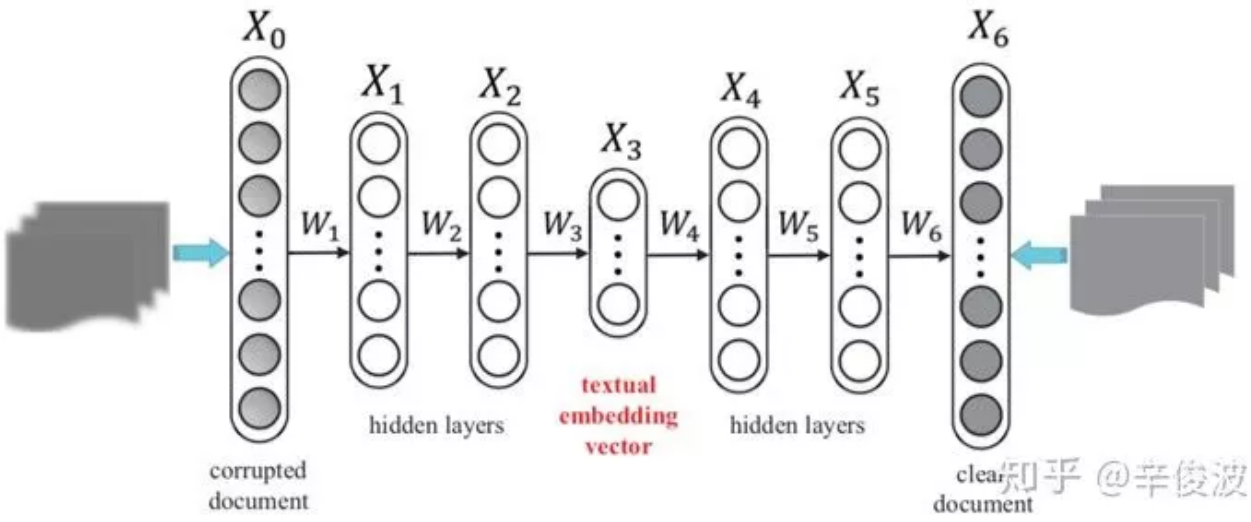


图3.12 textual embedding框架

c) 视觉特征Visual embedding: stacked convolutional auto-encoders (SCAE)

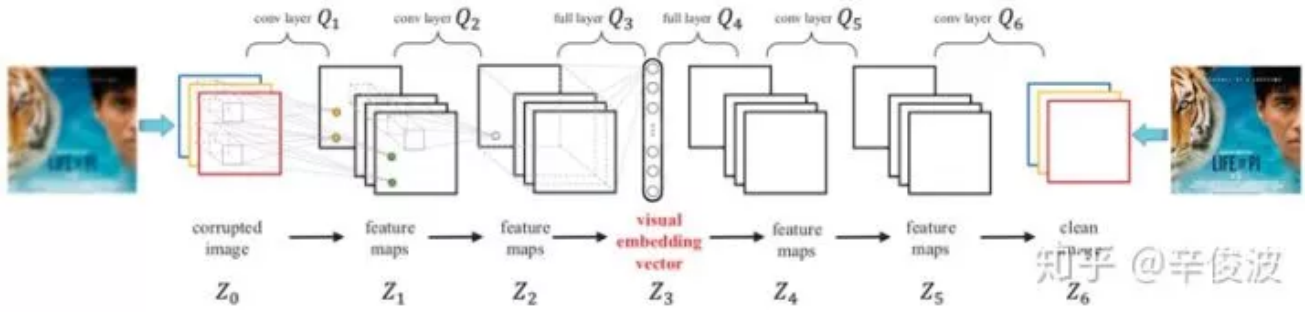


图3.13 visual embedding框架

(3) matching function

得到用户向量和item向量后，用向量点击表示user和item的匹配分数；损失函数则用如下的pair-wise loss表示

$$p(j > j'; i|\theta) = \sigma(\mathbf{u}_i^T \mathbf{e}_j - \mathbf{u}_i^T \mathbf{e}_{j'})$$

Part3 基于representation的深度匹配方法总结

微观层面

总结上述基于CF的方法，可以用如下的范式作为表达

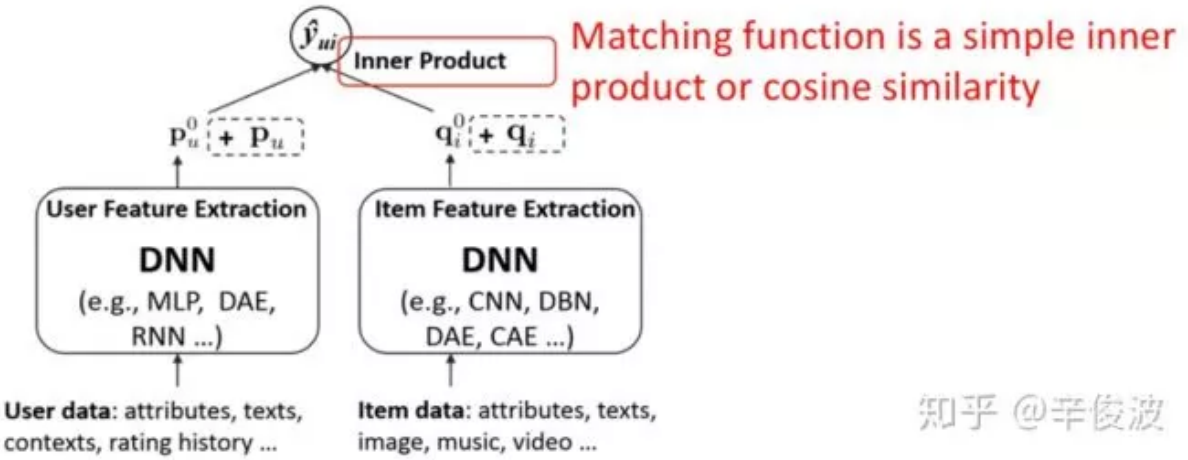


图3.14 基于CF的深度匹配模型范式

（1）representation learning: 目的是学习到user和item各自的representation(也叫latent vector, 或者embedding)

（2） 特征表达：user侧特征除了用户id本身userid，可以加上其他side info；item侧特征除了物品id本身itemid, 还有其他文本特征、图文特征、视频帧特征等信息

（3） 模型表达：除了传统的DNN，其他结构如Auto-Encoder（AE）, Denoise-Auto-Encoder(DAE)，CNN，RNN等。

基于representation learning的深度匹配模型不是一个end-2-end模型，通过user和item各自的representation作为中间产物，解释性较好，而且可以用在出了排序阶段以外的其他环节，例如求物品最相似的item集合，召回环节等。

宏观层面

对于深度模型，主要分为基于representation learning的深度模型以及match function learning的深度模型。基于representation learning的深度模型学习的是用户和物品的表示，然后通过匹配函数来计算，这里重点在与representation learning阶段，可以通过CNN网络，auto-encoder, 知识图谱等模型结构来学习。

整理本篇综述主要基于原始slides，对其中的paper部分粗读部分精读，收获颇多，在全文用如何做好推荐match的思路，将各种方法尽可能串到一起，主要体现背后一致的思想指导。多有错漏，欢迎批评指出。

Part4 参考文献