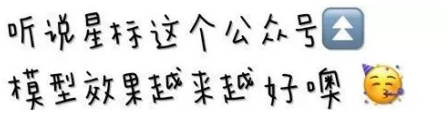
重磅整理！推荐系统之深度召回模型综述（PART I）

原创 一块小蛋糕 NewBeeNLP 2020-10-29

收录于话题
#推荐搜索

7个



> NewBeeNLP原创出品
> 公众号专栏作者@**一块小蛋糕**
> 知乎 | 推荐系统小筑

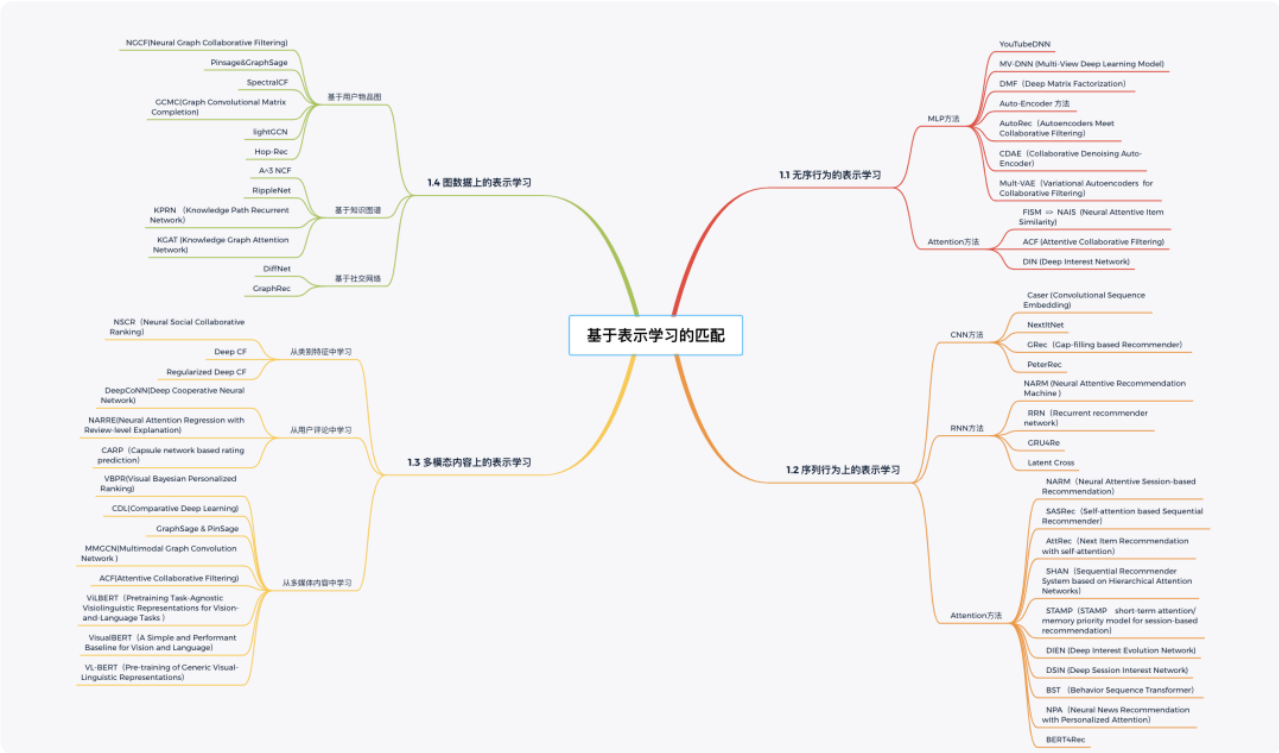
最近读完了李航、何向南的《**Deep learning for matching in search and Recommendation**》，文章思路清晰，总结详实到位，值得一再翻阅，就想借这篇文章结合自己最近一年多的推荐召回工作内容，总结一下推荐系统中的深度召回模型，论文因篇幅限制，很多模型并未详细介绍，因此本文补充了一些内容。

> 这篇综述文章现在好像不好下载，很多同学私信我，一个个发邮箱不太方便，现在大家可以直接在NewBeeNLP公众号后台回复『**DLM**』下载。

匹配（matching）是衡量用户对物品的兴趣的过程，也是推荐召回中的工作内容。机器学习中是以learning to match的方式根据输入表示和标记数据学习一个匹配函数。而深度学习在其发展过程中以强大的表示学习和泛化能力加上算力提升、数据规模暴涨都使得深度模型在推荐召回中大放异彩。

本系列文章的总结思路是将推荐中的深度召回模型根据学习内容分为两大类：**「表示学习类」**和**「匹配函数学习类」**。

- 表示学习类召回模型中根据输入数据的形式和数据属性又可以分为**无序交互行为类**、**序列化交互行为类**、**多模态内容类**和**连接图类**。



- 匹配函数学习类模型则包括**双路匹配函数**和**多路匹配函数**的学习。

由于篇幅限制，本系列将分成多篇文章分享，欢迎持续关注！👻

1 基于表示学习的匹配

推荐中的召回的最大挑战是不同空间的不匹配问题，而隐空间可以很好的解决这个问题，即将用户和物品都映射到同一个可以比较的低维空间内。这里表示学习的匹配模型所使用的就是隐空间模型框架。表示学习的过程就是学习两个函数：$\phi_u(u), \phi_i(i)$将用户和物品映射到一个新空间中，则用户u和物品i的匹配模型：$f(u,i) = F(\phi_u(u), \phi_i(i))$，F是内积或Cosine之类的相似度函数。

不同的神经网络可以实现不同的表示函数ϕ_u和ϕ_i，根据模型输入数据的形式和数据属性将表示学习进一步分为4类：无序交互行为类、序列化交互行为类、多模态内容类和连接图类。

1.1 无序行为的表示学习

首先是无序行为上的表示学习，传统的矩阵分解以用户的交互历史表示用户，即每一维代表一个物品的multi-hot向量，形成打分矩阵，利用one-hot 的ID向量再经过一层线性映射得到用户和物品的表示。而在深度学习中则是依靠深度模型学习到用户和物品的表示，常用的方法有**MLP方法**、**Auto-Encoder方法**和**Attention方法**。

MLP方法

YouTubeDNN

2016年YouTube提出的深度推荐模型，将每个类别特征都映射为一个Embedding向量，像观看过的视频、搜索过的词项这种序列特征使用平均池化得到序列Embedding；将所有Embedding和连续特征拼接后送入三层MLP得到最终打分。

认为MLP可以发挥其近似任何连续函数的优势学到特征Embedding间的交叉，但这种模型是把特征交叉信息编码进MLP的隐藏单元，并不能显式的区分出哪个交叉对预测更重要。而且2018年Beutel等人证明MLP实际上很难学习到乘法操作，而乘法操作是捕获特征交叉信息的重要方式。

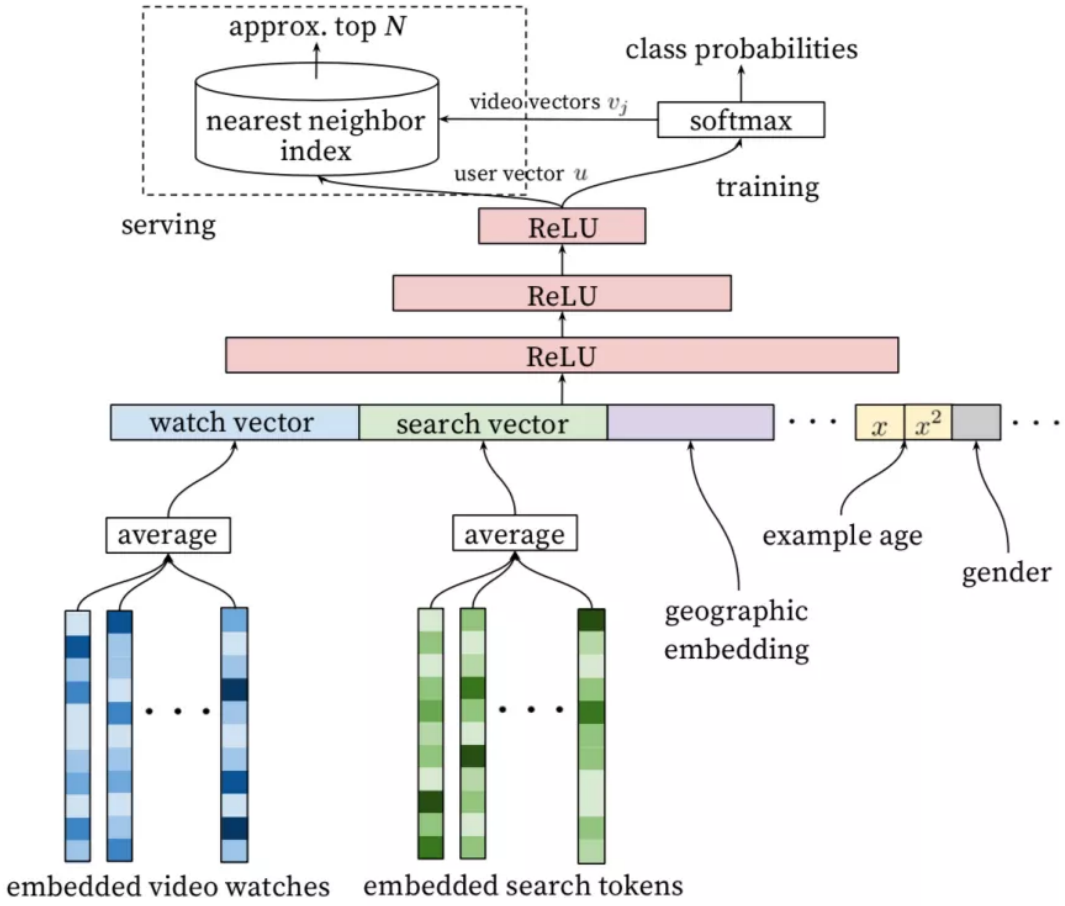


Figure 3: Deep candidate generation model architecture showing embedded sparse features concatenated with dense features. Embeddings are averaged before concatenation to transform variable sized bags of sparse IDs into fixed-width vectors suitable for input to the hidden layers. All hidden layers are fully connected. In training, a cross-entropy loss is minimized with gradient descent on the output of the sampled softmax. At serving, an approximate nearest neighbor lookup is performed to generate hundreds of candidate video recommendations.

MV-DNN (Multi-View Deep Learning Model)

2016年微软的MV-DNN是基于DSSM（双塔）的跨领域（多塔）构建用户表示的模型。模型提出假设：**在一个领域内相似的用户在另一个领域内也相似**，比如app下载领域相似的用户可能也具有相似的文章阅读喜好。MV-DNN可以很好的将user和item的大量特征编码到隐语义空间，通过兴趣匹配模型学习从user到item的映射关系。既解决了新用户的冷启动问题，同时由于以大量用户行为做特征，利用跨多个域的行为来补充用户信息，对用户兴趣表示也更加精准。该方法对新用户推荐提升明显。

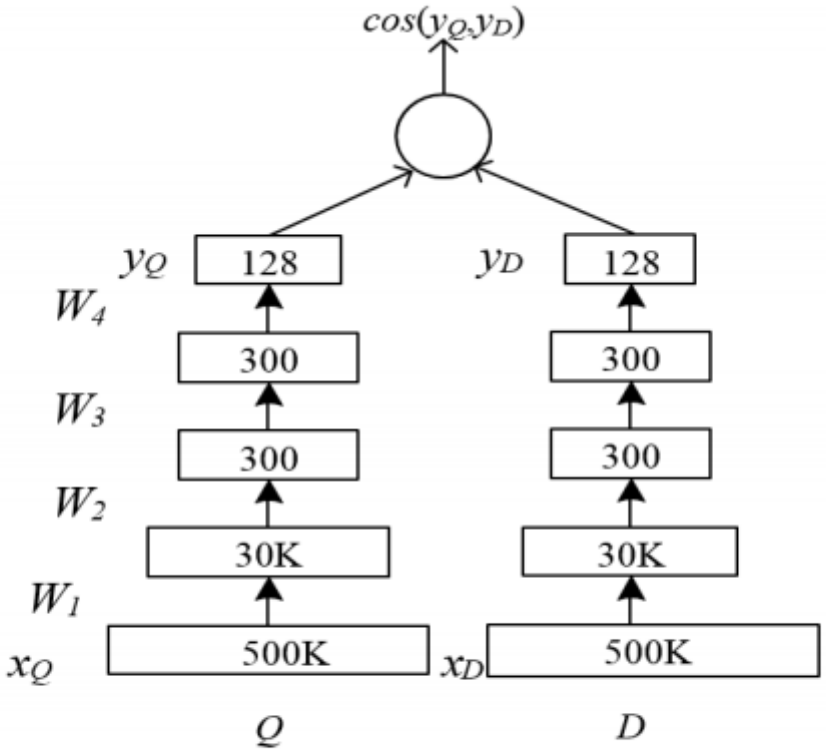
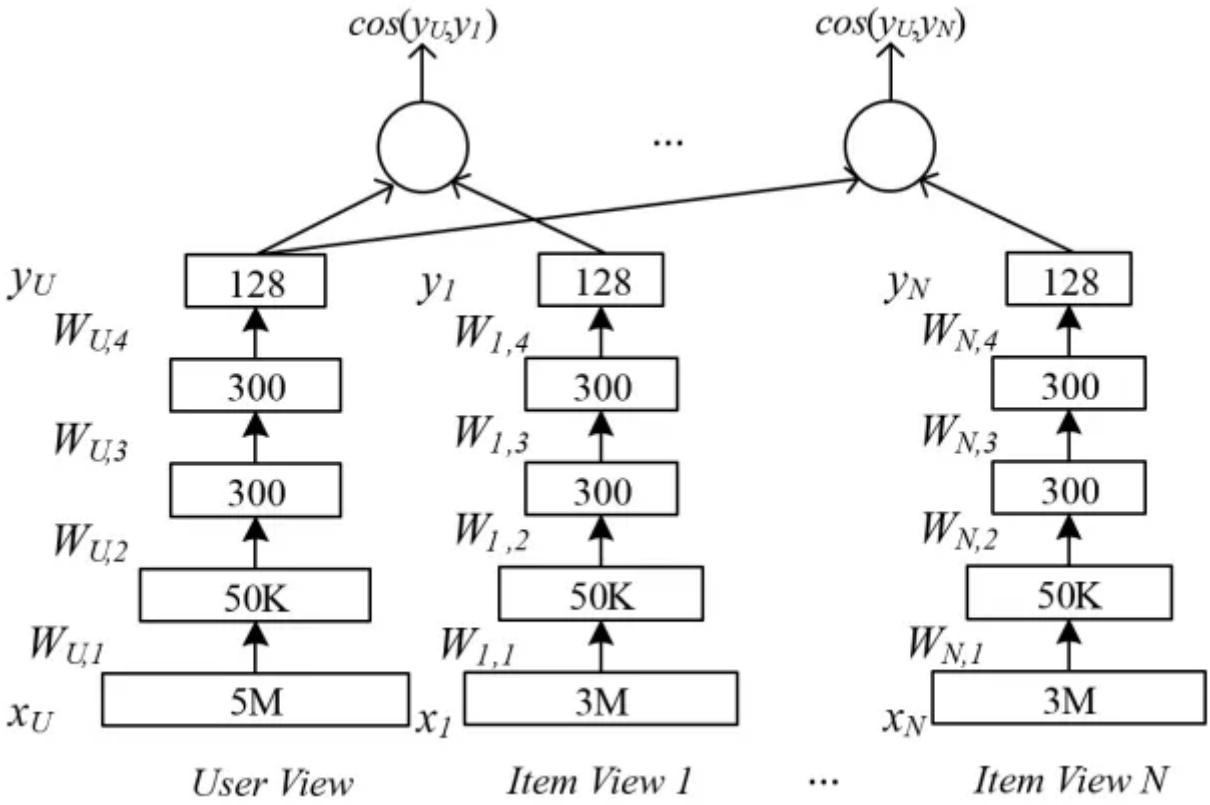


Figure 1: The illustration of the deep structured semantic model (DSSM).



由于深度模型需要处理大规模训练数据的大规模特征，文中提出了几种可行的降维方法：

- **「top 特征」**：选择最频繁的topk个feature，并且用tf-idf过滤掉停用词特征

- **「k-means」**：将特征聚类，相似特征聚到同一个类簇中，并将类簇信息表示为新的特征Y。原有特征为N个，聚簇个数为K，特征的维度将从所有特征o(N)降维到类簇个数o(K)。特征处理阶段，新特征Y(y1,y2,...,yi,...,yk-1,yk)具有K个维度，每个维度的值为属于该类簇的特征出现次数加和，最后对Y进行归一化处理。合适的类簇个数对特征表达能力十分重要。小的类簇个数会导致非常多的内容聚簇在一起，从而导致特征被稀释。文中的特征数有3.5M，尝试聚簇个数10K个，即平均每个类簇包含350个特征
- **「LSH」**：通过一个随机矩阵将输入特征映射到一个低纬度向量表示，同时在新的空间中保持pairwise cos距离关系。为了保证准确性，文中设定Y的维度k=10000,和k-means维度相同。由于LSH的k个维度是随机映射，彼此相互独立，很适合进行并行计算。
- **「缩减训练样本规模」**：压缩训练样例，每个用户的训练样例数只压缩为一个。文中具体做法是将同一用户所有训练样本的各维度特征进行分数平均，最终一个用户得到一个训练样本，从而减小user-item pair。由于训练样本的表示变化，评估方式也会变化。目标函数将改为最大化用户特征和平均特征的相似度

DMF (Deep Matrix Factorization)

2017年的DeepMF从用户对物品的评分矩阵中直接构建交互矩阵，采用DSSM的双塔结构，每个塔都用MLP从交互矩阵的multi-hot向量中学习到对应的表示。即

$$p_u = MLP_1(y_{u*}), q_i = MLP_2(y_{*i}), f(u,i) = cosine(p_u, q_i) = \frac{p_u^T q_i}{||p_u||^2 ||q_i||^2}$$

模型的损失函数使用了规范化的交叉熵损失:

$$L = -\sum_{(i,j)\in Y^+ \bigcup Y^-} (\frac{Y_{ij}}{max(R)} log\hat{Y}_{ij} + (1 - \frac{Y_{ij}}{max(R)}) log(1 - \hat{Y}_{ij}))$$

用max(R)即评分矩阵中的最大值对评分值做规范化。

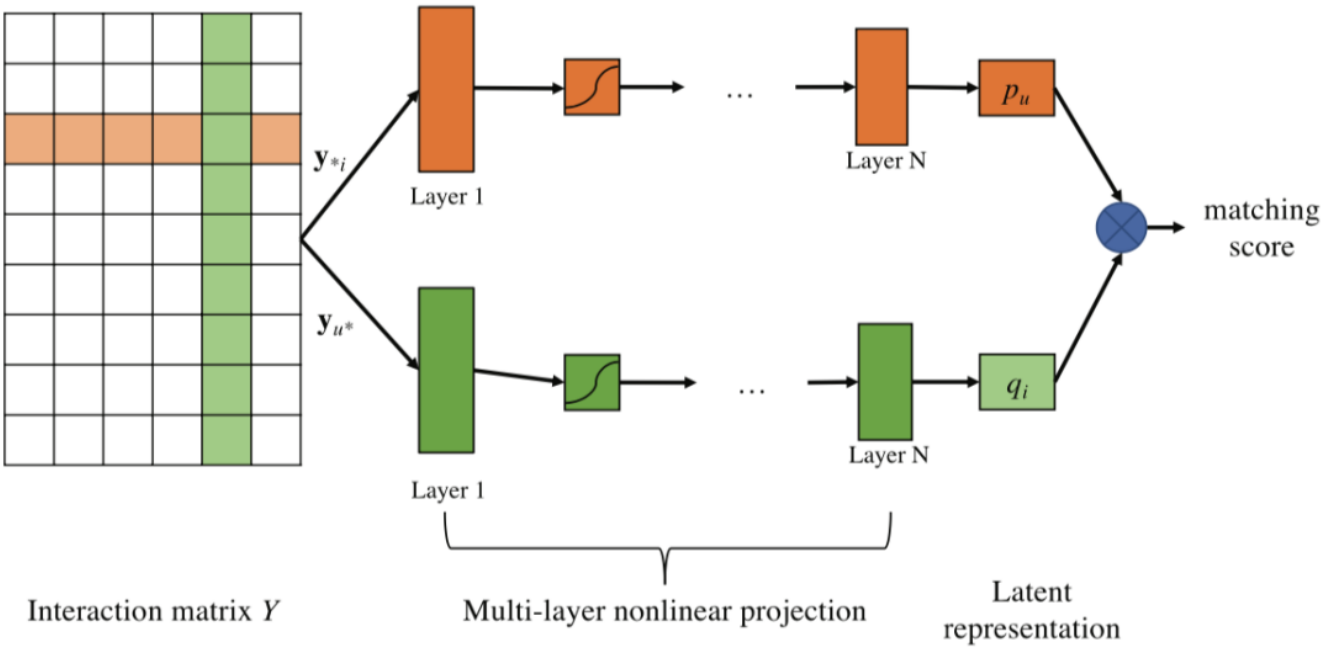


Figure 5.1: Model architecture of DeepMF.

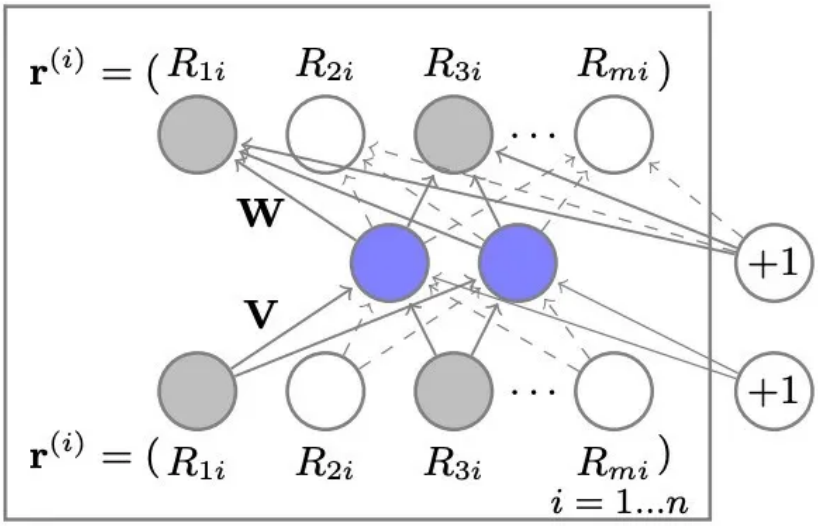
Auto-Encoder 方法

Auto-Encoder是个单隐层神经网络，主要利用其隐层学习低维特征表示或者利用重建层填充交互矩阵的空白值。

AutoRec（Autoencoders Meet Collaborative Filtering）

15年的AutoRec是基于Auto-Encoder模式学习CF模型，对比AutoRec原论文发现在这篇survey中P110关于AutoRec的描述其实是基于用户的AutoRec，但本文中说的是item-based AutoRec。基于用户的AutoRec输入数据是用户u对所有物品的打分向量y_{u*}，评分重建过程：$\hat{y}_{u*} = \sigma_2(W\sigma_1(Vy_{u*} + b1) + b2)$，重建之后的向量$\hat{y}_{u*}$即预测用户u与所有物品的匹配分值的向量，AutoRec的目标函数使用的是RMSE损失：$L = \sum_{u=1}^{M} ||y_{u*} - \hat{y}_{u*}||^2 + \lambda||\theta||$。这里也可以使用其他的损失函数如交叉熵损失、hinge损失和pairwise损失等。

原论文中是以item-based AutoRec为例讲解的，如下图所示，输入数据是所有用户对物品i的评分向量y_{*i}，其他部分与上面一样。文中的结论是item-based AutoRec效果好于user-based AutoRec，因为物品i的评分数一般比用户u的打分数多，而且用户向量方差更大，不容易学习。

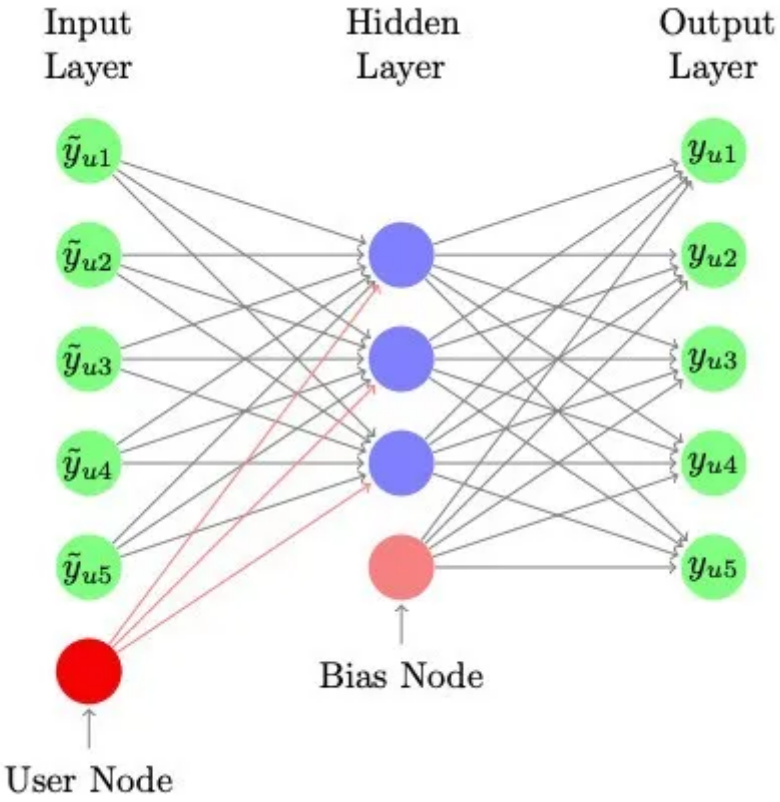


其实可以将基于用户的AutoRec看做是用MLP在用户交互历史上学习用户表示，用Embedding层学习物品表示，即：$f(u,i) = \hat{y}_{u*,i} = \sigma_2(w_{i*}\sigma_1(Vy_{u*} + b1) + b2)$，其中$w_{i*}$表示W的第i行，可以看做是物品i的id对应的Embedding，$p_u = MLP(y_{u*}) = \sigma_1(Vy_{u*} + b1)$是用户u的交互向量经过一层MLP的输出，匹配分值则是用户的表示p_u和物品id的Embedding q_i的内积，这和隐空间模型的定义是一致的。如果Auto-Rec中用到多层隐层形成深度Auto-Encoder，可以看做是用多层MLP学习用户表示，因此auto-encode结构也可以看做是DeepMF的简化版。

CDAE (Collaborative Denoising Auto-Encoder)

CDAE是AutoRec的改进版本，使用隐式反馈数据对用户的偏好建模做topN推荐。CDAE以用户u的所有评分作为输入(即user-based autoRec模型)通过一层神经网络编码得到用户的隐藏表示，再通过一层神经网络还原用户的交互行为(隐式反馈)。

与简单的item-based auto-Rec不同在于CDAE在编码得到隐藏表示时加入了用户特征，语义上更丰富。同时为了使模型更具鲁棒性，CDAE对输入的用户评分向量引入随机噪声（可通过maskout或dropout或增加高斯噪声实现）防止模型学习到相等函数。模型结构如下图所示，输入层有I+1个节点，I是所有的物品数，还有一个是用户u的side information；中间隐层有K个节点，最后一个是bias，与输入层是全连接，将输入映射到低维空间中得到低维表示；输出层则是将低维表示重新映射到原始输入空间，通过最小化重建损失和参数的L2正则学习模型参数，同时使用负采样提高训练效率。



Mult-VAE（Variational Autoencoders for Collaborative Filtering）

2018年Netflix基于隐式反馈数据，提出Mult-VAE模型使用多项式似然变分自编码器解决变分推断用于推荐时参数过多的问题。推导过程比较复杂，篇幅限制，感兴趣的同学可以参考论文。

Attention方法

FISM => NAIS（Neural Attentive Item Similarity）

首先要了解2013年的FISM模型，即Factored Item Similarity Model，它是针对CF模型中只利用了用户物品交互信息且只使用userid表示用户的场景下，提出利用用户交互过的物品表示用户，提升用户表达的准确性，即$\hat{y}_{ui} = (\sum_{j \in R_u} q_j)^T v_i$，用用户喜欢过的所有item的累加和作为用户的表示，而目标物品的隐向量v_i是另一套表示，最终用向量内积表示相似度。

2018年何向南发现在学习用户表示的过程中历史交互物品对用户表示贡献的权重并不全是相同的，因此提出NAIS模型利用一个attention网络学习每个物品的权重。与FISM同样，每个物品关联两个Embeddingp_i和v_i分别表示物品作为目标物品和历史交互物品时的向量。

NAIS的匹配函数：$f(u,i)=(\sum_{j\in y_u \backslash i} a_{ij}q_j)^T v_i, a_{ij}=\frac{exp(g(v_i,q_j))}{[\sum_{j\in y_u \backslash i} exp(g(v_i,q_j))]^\beta}$，$a_{ij}$表示估计用户u与物品i的匹配分值时历史交互物品j的权重，计算a_{ij}时用到的β是取值$[0,1]$的平滑指数。注意力网络g可以用一个MLP实现，再由softmax函数对g的输出做平滑处理。注意力权重提升了表示学习的可解释性，而且能根据待匹配的目标物品的不同产生不同的用户向量。

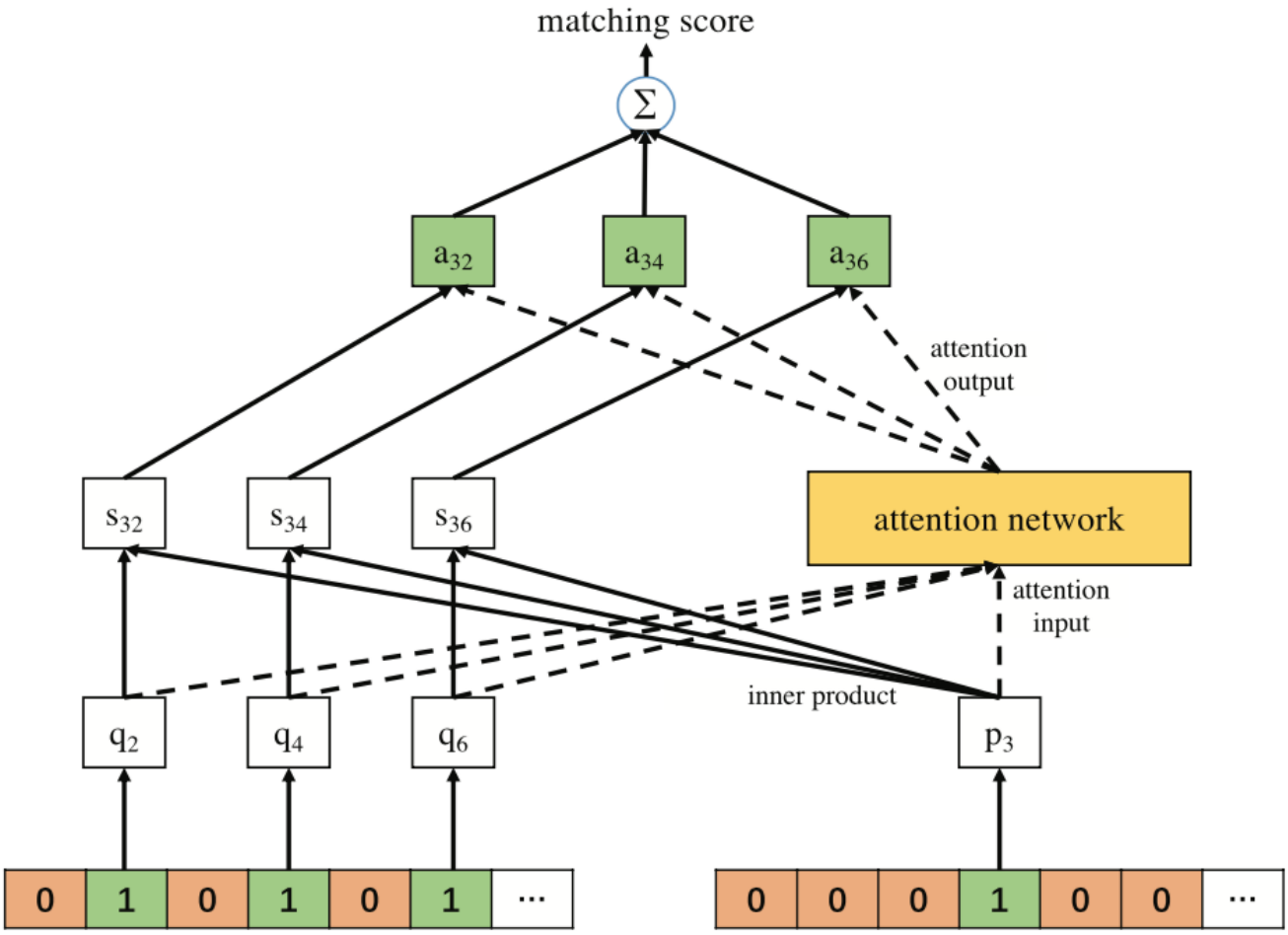


Figure 5.2: Model architecture of NAIS.

这里softmax增加这个平滑指数β的原因是作者在使用正常的注意力权重计算方式时，发现互动历史长的用户的权重会偏低，导致模型效果不佳，而增加一个β指数虽然会破坏注意力网络的概率解释，但确实使模型效果更好了。

作者还指出同时训练注意力网络和物品的Embedding层会减慢收敛速度，比较好的方式是先用FISM预训练好item的Embedding，导入NAIS模型后直接训练注意力网络，不仅收敛更快，模型效果也更好。

ACF (Attentive Collaborative Filtering)

传统的CF算法只利用了用户-物品交互矩阵信息，对用户的交互历史中的所有物品都视作相同权重，而且并不能很好的用于推荐多媒体信息上面。因此2017年新加坡国立大学提出基于两层注意力权重的协同过滤模型(Attentive Collaborative Filtering)，认为

在多媒体推荐领域，存在两层注意力：用户对每个推荐项(item，如音视频、图片等)的注意力；用户对某个推荐项的各个部分(component，如视频的帧、图片的区域)的注意力。

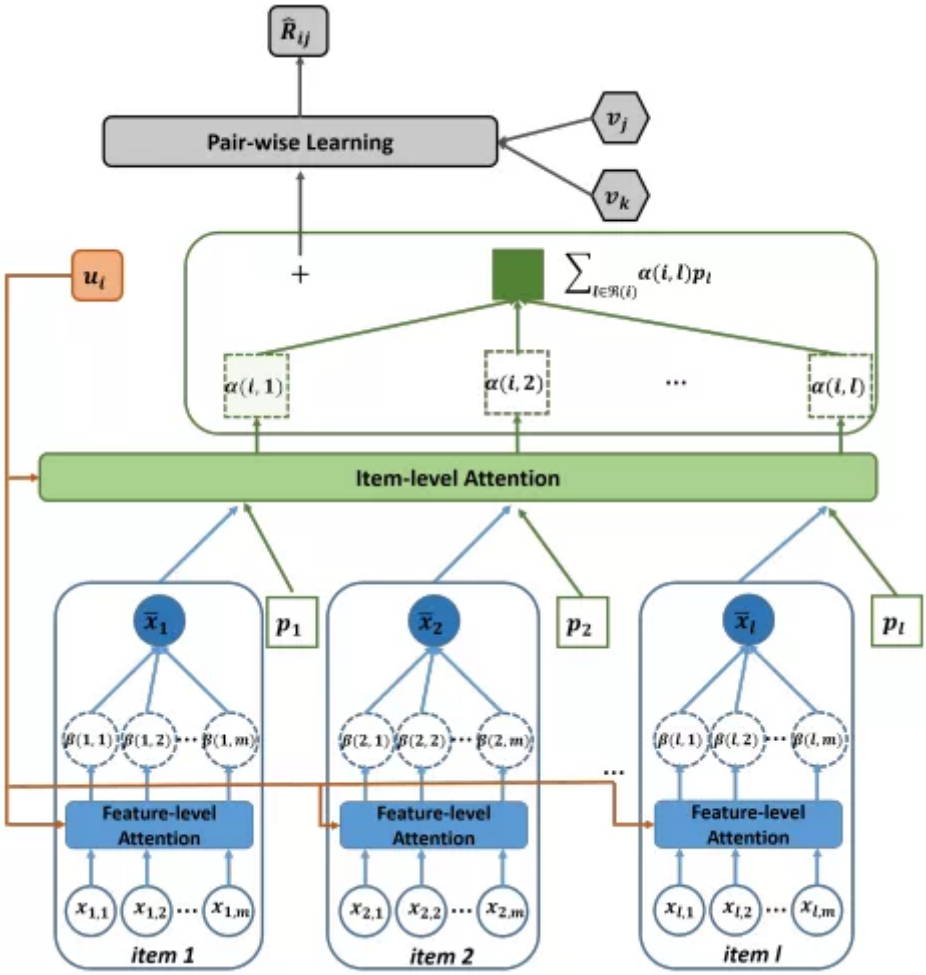
两层注意力分别用于学习用户对每个推荐项的的喜好和用户对推荐项的每个部分的喜好，然后用各部分的注意力权重和表示推荐项物品，各推荐项的注意力权重和表示对应的用户，即用户的表示为：$u_i + \sum_{l \in R(i)} \alpha(i,l) p_l$，$\alpha(i,l)$是推荐项的注意力权重。ACF模型优化的是BPR的pairwise目标函数是：

$$argmin_{U,V,P,\Theta} \sum_{(i,j,k) \in R_B} -ln\sigma(u_i + \sum_{l \in R(i)} \alpha(i,l) p_l)^T v_j - (u_i + \sum_{l \in R(i)} \alpha(i,l) p_l)^T v_k -$$

这里每个推荐项有两个因子向量，一个是隐空间的物品向量v_l，另一个是辅助物品向量p_l，预测用户和推荐项的匹配分值

$$\hat{R}_{ij} = (u_i + \sum_{l \in R(i)} \alpha(i,l) p_l)^T v_j \$ \text{，展开后：} \ \$\hat{R}_{ij} = u_i^T v_j + \sum_{l \in R(i)} \alpha(i,l) p_l^T v_j$$

第一部分是隐空间模型，第二部分是近邻CF模型。



上图是ACF的模型架构，首先从用户i喜欢的物品集合开始，每个物品l都有一个组件特征集合x_{lm}，比如图片的第m个空间位置、视频的第m帧。然后是两层全连接的组件级注意力子网络，以用户隐向量u_i和特征x_{lm}为输入，计算第m个组件的component级注意力权重$\beta(l,m)$:

$$b(i,l,m) = w_2^T \phi(W_{2u}u_i + W_{2x}x_{lm} + b_2) + c_2\$ \ , \quad \$\beta(i,l,m) = \frac{exp(b(i,l,m)}{\sum_{n=1}^{|x_{l*}|} exp(b(i,l,}$$

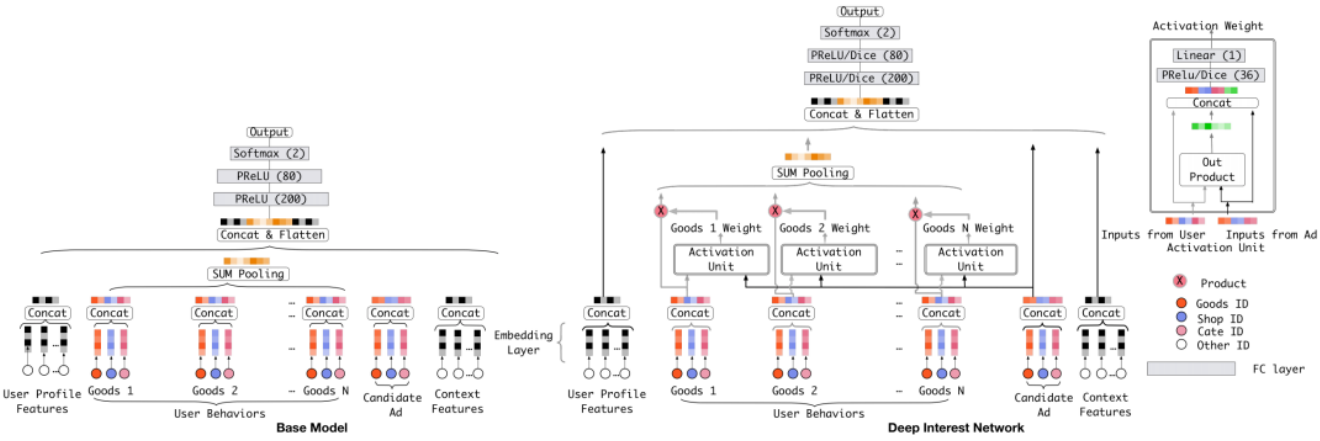
得到物品l内容表示：$\overline{x}_l = \sum \beta(l,m)x_{lm}$；得到物品内容表示后，接入两层全连接的item级注意力子网络，以用户隐向量u_i、物品隐向量v_l、物品的辅助隐向量p_l和内容特征$\overline{x}_l$计算每个物品的item级注意力权重$\alpha(i,l)$:

$$a(i,l) = w_1^T \phi(W_{1u}u_i + W_{1v}v_l + W_{1p}p_l + W_{1x}\overline{x}_l + b_1) + c_1\$ \ , \quad \$\alpha(i,l) = \frac{e}{\sum_{n\in I}}$$

得到用户i最终的近邻向量$\sum \alpha(i,l)p_l$；之后就是计算BPR损失的过程。ACF中使用DeepCNN的ResNet-152结构提取图片和视频帧的特征。

DIN (Deep Interest Network)

2018年9月阿里巴巴认为直接对用户行为历史做sum-pooling或average-pooling不能体现用户兴趣的多样性，因此提出DIN引入local-activation，以待预测物品与用户历史行为的相关性做权重来动态生成用户表示。



上图左侧是原有的深度推荐网络(Base Model)，通过sum-pooling用户行为历史的方式生成用户表示，而右侧DIN模型中将这部分替换成由activation unit层生成权重再求加权和的方式生成用户表示。这个activation unit层的结构和常规Attention的操作不完全一致，这里是将user的embedding和广告商品的embedding及二者外积拼接后接入Dice激活函数，再经过一层线性层得到activation权重，没有softmax操作，即不限

制注意力分值加和为1，论文中的观点是这样能保持用户兴趣的强度。这里作者还有三个创新点：

- **「Dice激活函数」**：由PReLU函数改进而来，原本PReLU：$f(s)=p(s)*s+(1-p(s))*\alpha*s, p(s)=I(s>0)$，Dice中将p(s)泛化为$p(s)=\frac{1}{1+e^{-\frac{s-E[s]}{\sqrt{Var[s]+\epsilon}}}}$，$E[s], Var[s]$是batch内数据的均值和方差。
- 以离线指标GAUC代替线上指标：$AUC=\frac{\sum_{i=1}^{n}\#impression_i*AUC_i}{\sum_{i=1}^{n}\#impression_i}$
- **「batch内正则」**：解决大规模稀疏场景下（比如用户或商品id空间都在千万级以上），使用SGD优化并引入L2正则后需计算全量参数的L2范数导致的复杂度过高的问题，将计算全量参数的L2范数改为只计算batch内用到的参数的l2范数，$L_2(W)=\sum_{j=1}^{K}\sum_{m=1}^{B}\sum_{(x,y)\in B_m}\frac{I(x_j!=0)}{n_j}||w_j||^2$，B表示batch数，$B_m$表示第m个batch。

下期预告