

# 深度学习技术在美图个性化推荐的应用实践

文章作者：陈文强、白杨、黄海勇

**导读：**美图秀秀社交化的推进过程中，沉淀了海量的优质内容和丰富的用户行为。推荐算法连接内容消费者和生产者，在促进平台的繁荣方面有着非常大的价值。本次分享探讨美图在内容社区推荐场景下应用深度学习技术提升点击率、关注转化率和人均时长等多目标的实践经验。

## \1. 美图社区个性化推荐场景概况与挑战

### \2. embedding 技术在召回阶段的应用实践

- 基于 Item2vec 的 item embedding
- YouTubeNet 和双塔 DNN 在个性化深度召回模型应用实践

### \3. 美图排序模型的研发落地

- NFWFM 模型研发迭代历程和经验
- 多任务学习 ( Multi-task NFWFM ) 在多目标预估场景的探索与实践

## ■ 美图社区个性化推荐场景与挑战

### 1. 业务场景

美图社区个性化推荐场景大大小小有十多个，其中流量比较大的场景是美图秀秀 app 的社区内容推荐 tab ( 图1 )，这个场景以双列瀑布流的形态给用户推荐他最感兴趣的内容。

当用户点击感兴趣的图片后会进入图1-2的相似推荐 feeds 流场景。在这个场景下，用户消费的图片和视频，都是和用户刚刚点击进来图片是具有多种相似性的，如视觉、文本、topic 等。而如果用户是从双列瀑布流里点击视频，则会进入到图1-3的视频 feeds 流场景。这个场景主打让用户有沉浸式的消费体验。以上是美图社区内容推荐的主要业务场景。

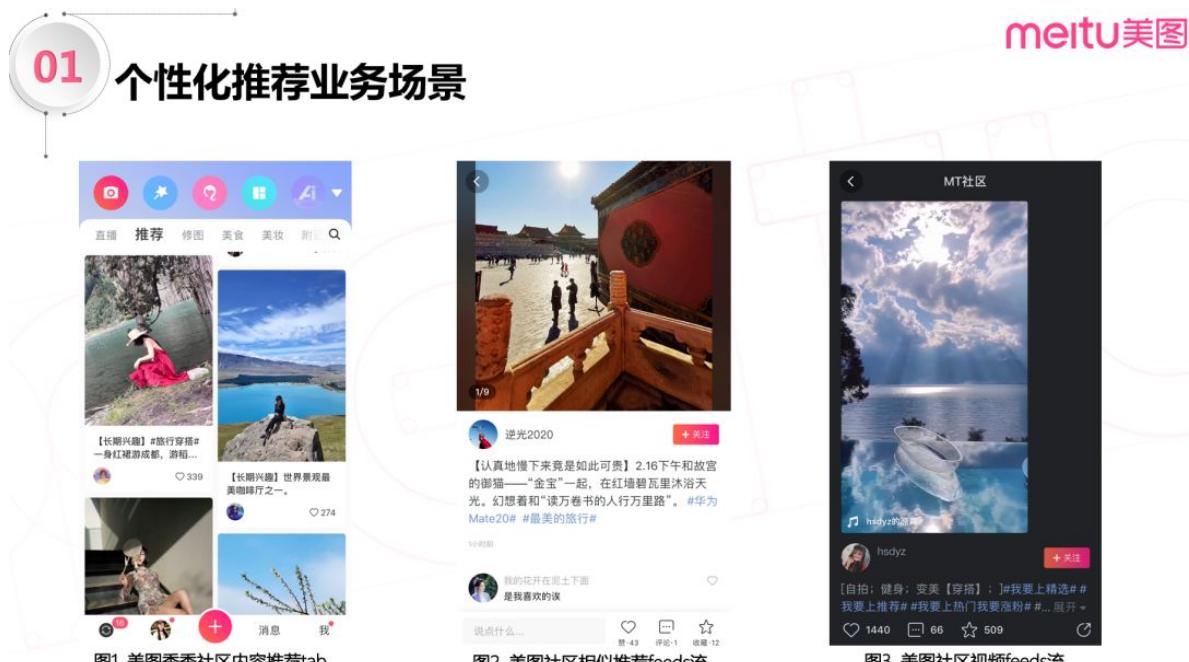


图1 美图个性化推荐业务场景

### 2. 工作目标

个性化推荐的首要目标是理解内容。从内容本身的视觉、文本以及特定场景下用户的行为来理解社区里可用于推荐的内容。接下来是理解用户，通过用户的基础画像（年龄，性别等）、设备画像（OS，机型等），以及用户的历史行为来挖掘其兴趣偏好。

再理解了社区的内容和用户之后，才是通过大规模的机器学习算法进行精准推荐，千人千面地连接用户与内容，从而持续提升用户体验，促进社区繁荣。



图2 美图个性化推荐业务目标

### 3. 挑战

在实际生产实践过程当中，主要遇到挑战如下：

- **海量**

美图内容社区有月活超过1亿的用户，每天有100多万的候选图片和视频。在这种数据规模下，只在小数据规模下验证有效的复杂算法模型无法直接应用到工业界线上系统。

- **实时**

算法需要在端到端小于 300ms 的时延里，每天处理超过3亿的个性化排序请求。这要求召回和排序算法不能过于复杂，要能够进行高效的计算。

- **长尾**

在实际场景中，用户分布以及 item 行为分布都是长尾的：

- 用户分布的长尾性：新用户占比超过27%；
- 曝光分布的长尾性：关注<<点击<<曝光。

在这样长尾数据上进行预估要求我们的模型具备稳定的泛化能力。

01

# 挑战

## 海量

1亿+ 用户

100万+ 候选图片/视频

## 实时

每天3亿+ 个性化排序请求

端到端请求时延在300ms以内

## 长尾

新用户: 曝光占比27%

Item分布: 曝光分布长尾, 关注<<点击<<曝光

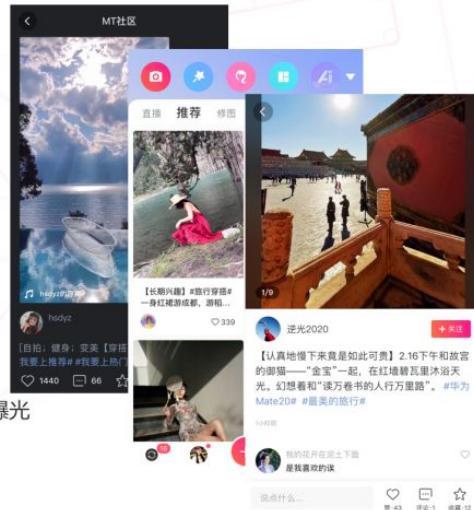


图3 美图个性化推荐的挑战

## 美图深度学习技术栈——召回端

在上述的应用场景和技术挑战下, 美图是如何将深度学习应用到个性化推荐中的召回端和排序端的呢? 我将会在下面为大家一一介绍。

召回端的 Item embedding 技术和召回模型, 用于从百万级别的候选集里挑选 TOP 500 个用户最感兴趣的候选集。相对于召回端, 排序端的深度排序模型能够融合多种召回来源并进行统一排序, 排序模型能包容规模更大的细粒度特征, 相对召回模型, 排序模型能够实现更加精准的推荐。

美图目前部署在线上系统的召回技术主要包括 Item2vec, YouTubeNet, 以及双塔 DNN。

### 1. Item2vec

Item2vec, 是一种通过用户行为来理解内容的方式。

传统的理解内容方式是基于用户行为构造 item 侧的统计类特征, 例如 item 的点击率, 收藏率等。这些特征是非常有效的, 但是对内容的理解维度比较单一。

另一种方式是从图片本身的视觉来提取比如图片质量、清晰度、图片物体等等。还可以通过内容本身的文本特征, 比如关键词, 实体词等等, 来帮助理解内容。这些维度的特征在内容冷启动中是很有效的。但是他们无法表达内容的某些潜在特性, 比如某个内容是否给用户呈现出清新有趣的感觉。这种潜在的特征借助用户的行为来理解比较合适。Item2vec 正是这样一种技术, 它基于短时间内被浏览的 item 具有内在相似性的假设来学习 item 的 embedding。

在图4中可以看到 item2vec 在美图社区图片上的部分效果。可以看到和查询图在训练数据中高频共现和中频共现的 item, 在背景和主体人物上和查询词是高度相似的。而低频共现的部分和我们的查询图片有些差异, 不过主体内容总体上还是比较相似的。

总体而言, item2vec 是一种学习 item embedding 的成熟方案。



图4 item2vec 可视化效果

Item2vec 学习出来的 item 向量是美图多种向量检索式召回策略的底层数据，包括实时兴趣，短期兴趣等等。比如当用户点击了某个 item，系统会实时地通过向量内积运算查询相似的 item 并插入到召回源头部，用于下一轮的排序。

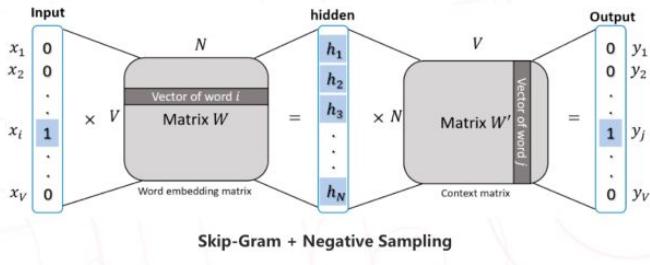
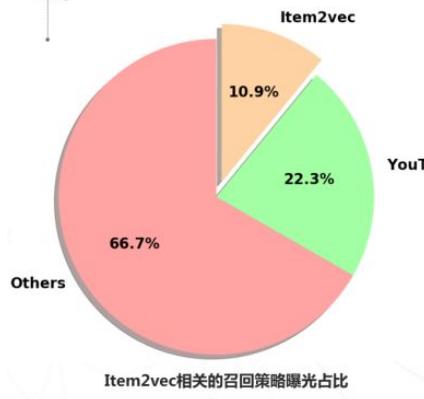
使用 item2vec 学习出来的 item 向量作为底层数据的召回策略，在美图应用非常广泛，它们在整体曝光中占了10%以上。

实际应用时，我们是基于 skip-gram+negative sampling 来做 item2vec 的。

它是一个只有一个隐层的深度学习模型。输入端是用户的点击序列，输出端是与输入端的 target item 邻近的64个 item。64相对于 NLP 里取的5-6个是比较大的，这是因为用户的点击序列不像自然语言那样具有严格的局部空间句法结构。在比较大上下文窗口中，更容易找到和目标 item 相似的上下文 item，模型更容易学习。

02

## 召回——Item2vec



| 关键参数               | 含义            | 考量          |
|--------------------|---------------|-------------|
| context_item=64    | 上下文item数取64   | 更容易找到相似item |
| Threshold=5        | 过滤序列长度不少于5的样本 | 覆盖99%的item  |
| Neg_sample=128     | 负样本数采样为64*2   | 效果和性能的折中    |
| Embedding_size=128 | 128           | 效果和性能的折中    |

Item2vec 超参数

问题：仅基于item的上下文共现的Embedding Vector缺乏精细的个性化

图5 item2vec 应用实践

负采样的个数是正样本数的两倍，这是一个效果和性能折中，在我们的应用中正负样本数1:2，能够在天级别模型更新情况下，取得比较好效果。隐层的 embedding size 取128可以得到比较好的效果，这个取值同样是效果和性能的折中。另外，我们过滤了点击序列长度小于5的样本，这样过滤之后，我们的点击序列能覆盖99%候选图片和视频。

Item2vec 是学习 item embedding 向量的一种非常好的方案，它也覆盖了美图多个推荐业务。但是它不直接考虑用户的个性化行为，只考虑了训练样本中 item 与 item 之间的局部共现关系。如果要利用上丰富的用户侧特征，实现个性化的话，那么我们需要借鉴其他方案，而 YouTubeNet 正是这样一种业界成熟的方案。

## 2. YouTubeNet

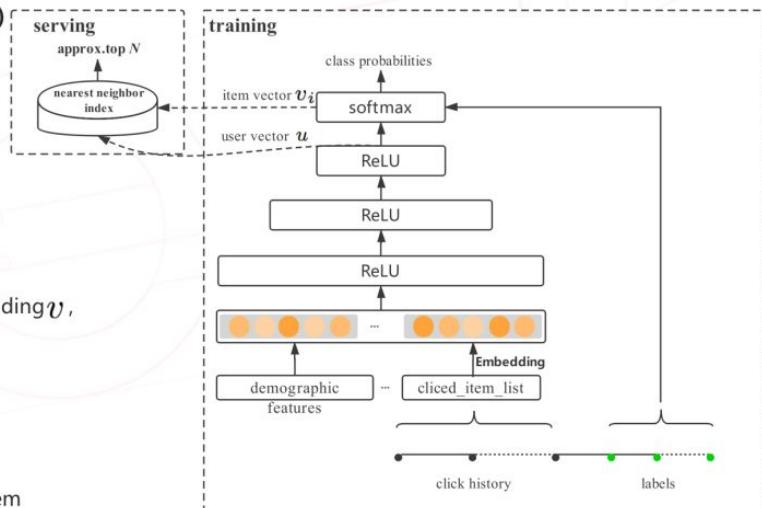
YouTubeNet 是 Google 于2016年提出的。与 item2vec 不同，YouTubeNet 在学习 item 向量的时候考虑了用户向量。从模型的优化目标上可以看出，是在给定用户向量的情况下，从候选池中筛选出该用户最感兴趣的 item 列表。

02

## 召回——YouTubeNet(1/2)

用户相关的item embedding(个性化)

$$P(v = i|U) = \frac{e^{v_i u}}{\sum_{j \in V} e^{v_j u}}$$



training阶段：

引入用户的个性化信息，

学习user embedding  $u$  & item embedding  $v$ ，

导出  $u, v$

serving阶段：

FAISS离线计算好每个用户的top N候选item

图6-1 YouTubeNet 应用实践

从上图右侧模型架构图可以看到，模型训练用的label是用户最近点击过的 item 列表，特征包括两部分，一部分是用户更早之前点击过的列表 ( clicked\_item\_list )，另一部分是用户的 demographic 统计特征，如年龄、性别等。引入上述用户的个性化信息之后，模型通过学习 user embedding 和 item embedding，并离线导出用户向量和 item 向量。线上使用时利用 FAISS 工具离线计算好每个用户的 top N 候选 item 集，提供给排序算法使用。不过这种离线存储候选集的方式，不能实时应对用户不断变化的兴趣，要捕获这种变化，需要实时采集用户不断变化的点击 item 数据，实时计算用户侧向量。

实时计算用户侧向量的工作，一共分了两部分：第一部分是离线部分，为下图右侧的虚线部分，这里模型一天一更新。离线部分基本流程和上一段所述相同，模型训练完之后导出 item 向量并在 FAISS 中构建好索引。第二部分是实时部分，这一部分借助 kafka，实时采集用户点击行为数据并构建 clicked\_item\_list 特征，接着请求离线训练好的模型，计算出用户侧向量，最后从 FAISS 中查询的候选集，输出给排序服务。

使用 YouTubeNet 模型实现实时计算用户侧向量之后，曝光占比22%的 YouTubeNet 给整体带来了点击率3.67%的提升，人均时长提升2.22%。

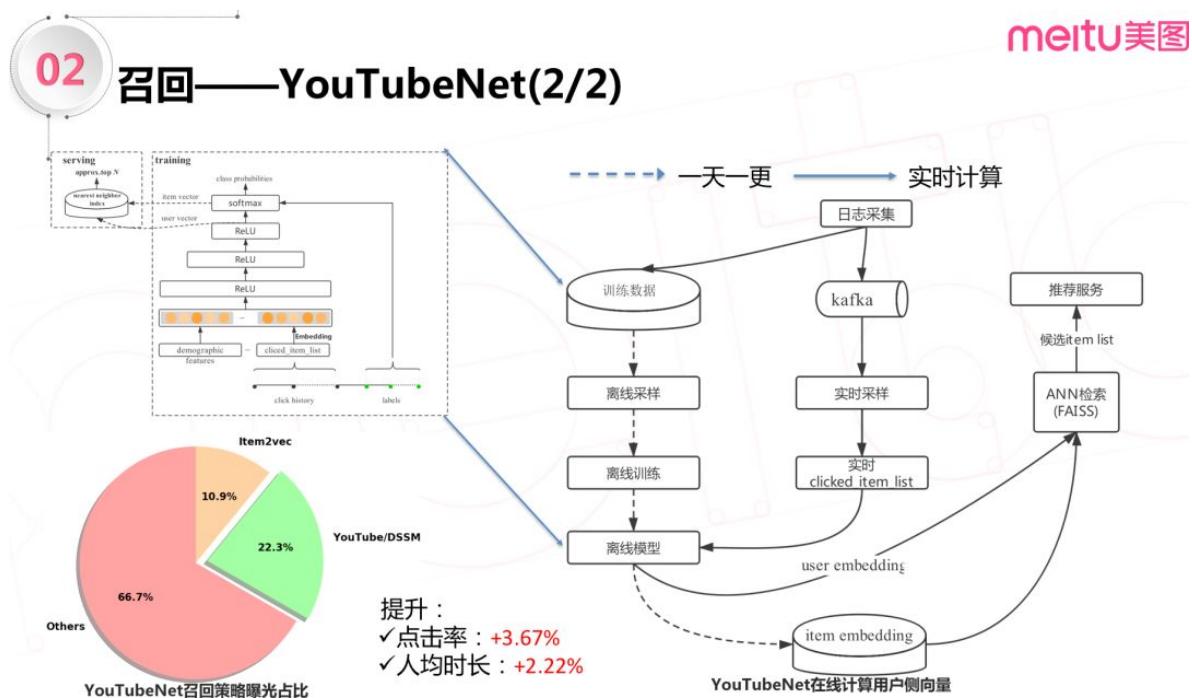


图6-2 YouTubeNet 应用实践

### 3. 双塔 DNN

双塔 DNN 模型，联合用户行为和 item 侧特征进行召回。双塔 DNN 模型构建用户侧 embedding 的方式和 YouTubeNet 是一样的：先给用户的点击行为序列，年龄性别等稀疏的特征做 embedding，再经过几个简单的全连接层，得到用户侧向量。对于 item 侧特征，双塔 DNN 引入另一个子网络来学习，学习方式和用户侧特征是一样的。

离线训练完了之后和 YouTubeNet 还是一样，把 item 向量提前导出并加载到 FAISS。在线上环境使用的时候，实时计算用户侧向量，来快速捕获用户兴趣。引入 item 侧特征，使得线上用户点击率提升 1.05%，人均时长提升0.76%。

02

## 召回——双塔DNN

联合用户行为和item侧特征进行召回

**training阶段:**

Item侧引入更多特征

**serving阶段:**

实时计算用户向量，

内积检索top N item

**效果：**

✓点击率：**+1.05%**

✓人均时长：**+0.76%**

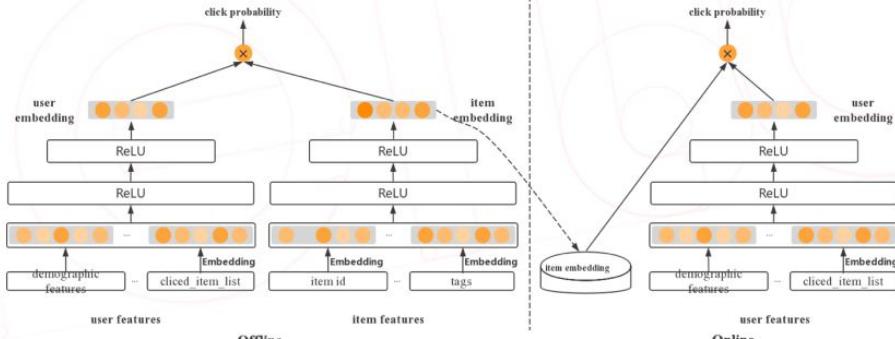


图7 双塔 DNN

### 4. 总结



Item2vec 学习出来的 item 向量具有很好的相似性，作为底层数据，服务了多个召回策略，在美图具有非常广泛的应用。包括实时兴趣，短期兴趣等等，覆盖了10%以上的曝光占比。YouTubeNet 和双塔 DNN 则分别引入用户侧和 item 侧特征，有监督地学习用户最感兴趣的 item 候选集，在美图个性化推荐召回层，累计点击率提升了4.72%，时长提升了2.98%。

### 美图深度学习技术栈——排序端

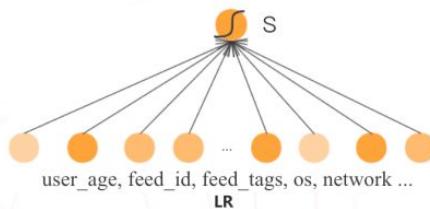
#### 1. 重新审视 NFM 模型

美图的第一代模型主打 LR 为主+人工特征组合。随着业务发展，大大小小的推荐场景越来越多，做特征的人力越来越紧张。恰逢深度学习在工业级推荐系统有大规模应用落地实践，因而逐渐将算法模型转向深度学习。

## 03

## 排序——深度学习时代

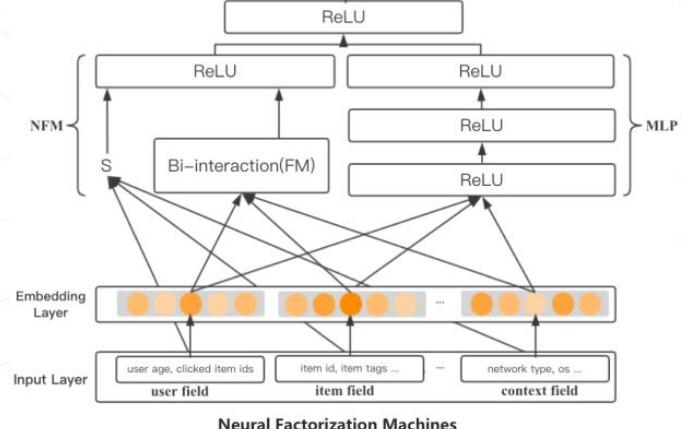
LR+人工特征组合：

**NFM模型：**

- FM&MLP端到端联合训练,
- 显式学习特征交叉

**效果：**

- ✓ 点击率：+5.53%
- ✓ 人均时长：+6.97%



[蒋文瑞 2018]. 深度模型DNN在个性化推荐场景中的应用

图8 NFM 模型

2018年年初，美图用 NFM 模型首次打败 LR 并取得稳定效果，NFM 模型创造性的将 FM 和深度模型端到端的联合训练，在底层就进行显示的特征交叉，NFM 论文的实验和美图数据集上的实验都表明了模型能够收敛更快也更加稳定。在美图的实践中，引入右侧多层感知机学习隐式的高阶特征交叉之后，效果进一步提升，好于原始的 NFM 模型。改进后的 NFM 模型在我们的推荐流场景中取得了5.5%的点击率，以及将近7%的时长提升。

在 NFM 模型取得稳定的正向效果之后，美图推荐团队继续探索了业界更多的模型。不过都未能落地，主要有以下两个原因。

第一个是像 Wide&Deep, DeepFM, DCN 等从模型的复杂度上看没有比 NFM 拥有更强的预估能力，计算效率也没有明显优势。离线评估和线上实验上都没有得到正向效果。

第二种情况是，xDeepFM 和 NFFM 离线指标提升了，但是计算复杂度很高。此外 NFFM 模型参数量大，内存是个瓶颈。导致它俩无法大规模落地。

| 模型名称       | 出处                 | 年份             | 实践效果(AUC)            | 优点                       | 不足                          |
|------------|--------------------|----------------|----------------------|--------------------------|-----------------------------|
| Wide&Deep  | Google             | 2016.06        | -0.024               | 首次公开&应用深度学习到推荐系统         | LR组件需人工组合特征，效率低             |
| DeepFM     | Huawei Noah        | 2017.03        | -0.003               | FM取代LR组件,不需人工组合          | 仅限二阶特征和隐式高阶特征               |
| <b>NFM</b> | <b>NUS</b>         | <b>2017.08</b> | <b>0.0(baseline)</b> | <b>引入特征的乘性关系到深度学习网络</b>  | 仅限于二阶特征和隐式高阶特征              |
| DCN        | Google             | 2017.08        | -0.002               | 特征组合阶数可控， <b>不限于二阶特征</b> | 低阶交叉无明显优势，高阶交叉时因多项式乘法，数值不稳定 |
| xDeepFM    | Microsoft Research | 2018.05        | +0.009               | 引入特征field信息构造复杂特征组合      | 计算复杂度高，落地困难                 |
| NFFM       | NJU                | 2019.04        | +0.009               | 引入特征field信息，构建二阶特征交叉     | 模型参数量大，落地困难                 |

经过一年多的探索，在业界的众多模型中我们没有找到合适美图推荐场景的排序算法。另外，在我们引入行为序列特征之后，NFM 的计算复杂度已经不能很好的支持线上流量。在这样的背景下，美图算法团队决定自行设计算法。

NFM 模型的优点在于，通过 Bi-Interaction，将 FM 模型端到端引入到深度模型，显式构建特征的乘性关系，加强模型预估能力，同时没有增加时间复杂度。但是在实际生产实践中，存在2个不足：

(1) NFM 需要足够的 Embedding layer 宽度来学习特征。在实际场景下，其宽度取200左右，效果最好。但是随着百万级别用户行为序列特征的加入，NFM 模型的计算量越来越大，越来越不能满足线上小于 300ms 的时延要求。

(2) 另一个不足是，NFM 模型本身存在 co-training 的问题，即：一个特征的学习，会不可避免地受到其他特征的影响。例如，用户的性别特征，与用户的网络环境特征是不相关的；但是 NFM 模型无法构建这种情况。

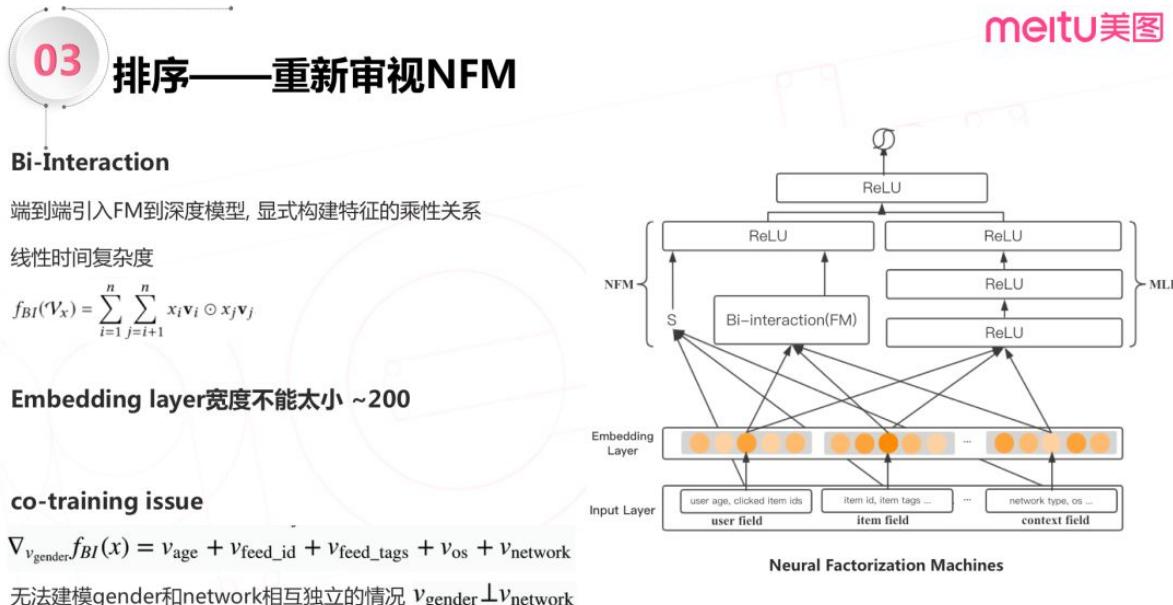


图9 NFM 模型的不足

基于以上不足，我们先来看下业界相关经验：



不管是在传统浅层模型时代还是在深度学习时代，引入特征的 field 信息之后，模型几乎是总能带来提升的。比如 FFM 仍然一直活跃在 Kaggle 等 CTR 预估比赛中，稳定的取得比不能建模 field 信息的 FM 更好的效果。而深度学习时代，业界很多公司比如2018年，微软离线验证 xDeepFM 引入特征的 field 信息之后，相对不能建模 field 信息的 DCN 同样取得了很明显的提升，即便在现在，xDeepFM 仍然是很优秀的模型。但是他们或者计算量太高或者参数量太大，导致无法大规模应用到线上系统。基于上面对 NFM 模型的优点的实验和分析，美图算法团队开始尝试 NFwFM 模型。

## 2. NFwFM 模型

### 2.1 模型整体架构

NFwFM 模型是在 FwFM 模型的基础上演化出来的：通过 Field-wise Bi-Interaction 组件，将 FwFM 引入到深度模型里面。

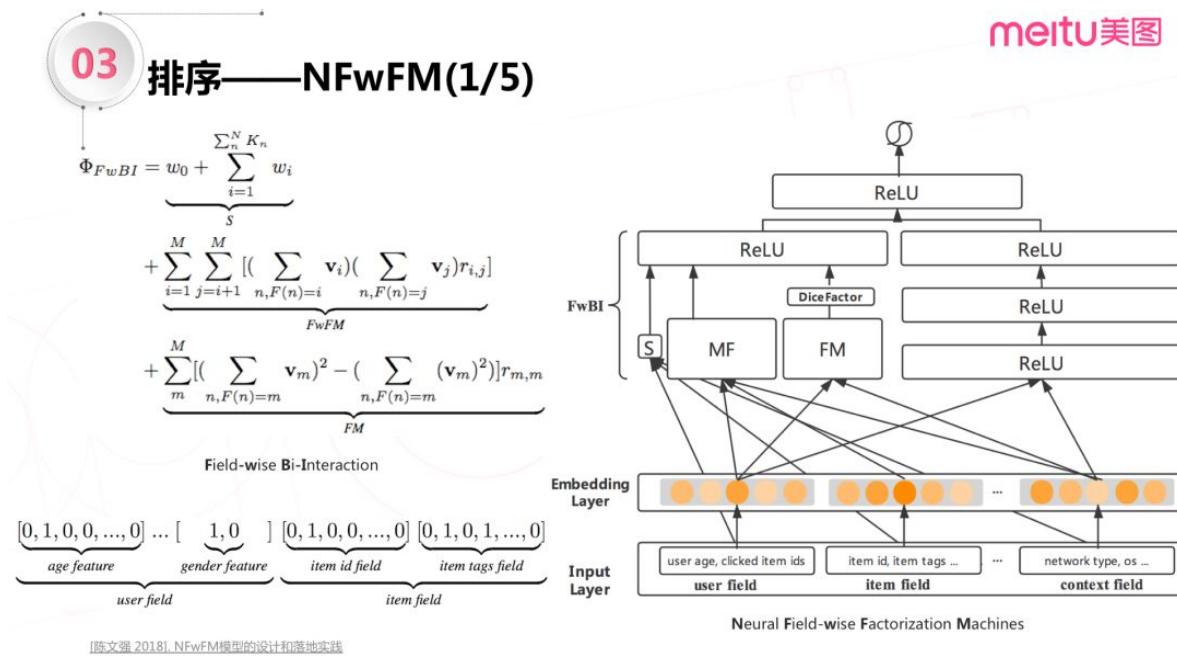


图10-1 NFwFM 模型整体架构

上图是 NFM 模型的整体架构，首先，把特征按照逻辑分为3个大模块：用户侧特征（包括年龄、性别等），item 侧特征（包括 item id, item 标签等）以及上下文侧特征。

接下来将 FwFM 分解成了3个子模块：第一个模块是线性加和模块（上图中 S 表示），不区别 field 学习的特征；第二个模块是矩阵分解模块（上图中的 MF 部分），用来学习 field 粒度下的特征组合，比如 user field 和 item field 的二阶交叉；第三个模块是 FM 模块，用来学习 field 内部 feature 粒度的特征组合。

## 2.2 FwFM 和 FFM 相比

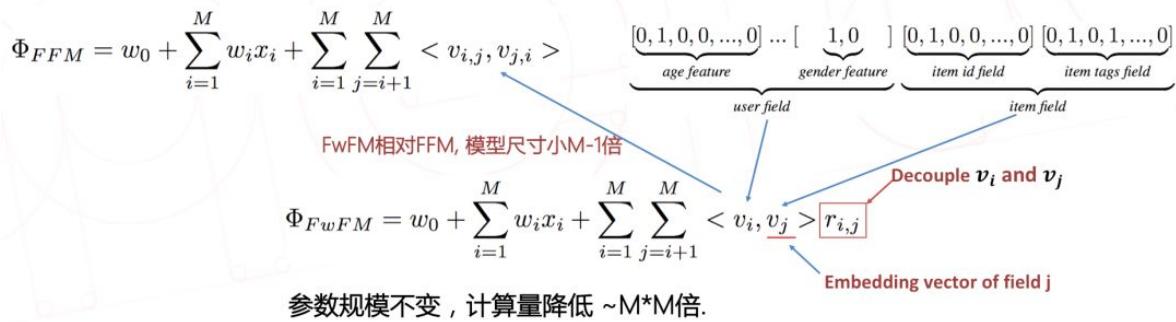
- (1) FwFM 模型尺寸相对 FFM 少  $M-1$  倍。在美图实际应用中，特征量大约可减少30倍。
  - (2) FwFM 模型引入了 Field 相关的权重  $r_{i,j}$ ，解决了 FM 存在的不相关特征在学习过程中互相干扰的问题。
  - (3) 离线评估显示，FwFM 模型的预测性能（例如 AUC 指标）和 FFM 基本一致，而参数规模大大降低。

FwFM 模型由于要建模 field 信息，导致它无法像 FM 那样具备良好的线性时间复杂度。因此，需要将 FwFM 做矩阵分解（上面架构图中的 MF 模块）。

## 2.3 MF 模块

如下图所示，模型分别从用户侧和 item 侧提取特征向量  $v_i$  和  $v_j$ ，在这两个向量上进行矩阵分解，用来学习 field 粒度的特征组合。实际应用中，需要分别对用户侧、item 侧、context 侧进行两两矩阵分解，因此共有3个矩阵分解子模块。

通过离线评估显示，MF 分解前后的 FwFM 模型，其 AUC 等指标持平，但是相同参数规模下计算量降低  $M \times M$  倍，计算效率大大提升。



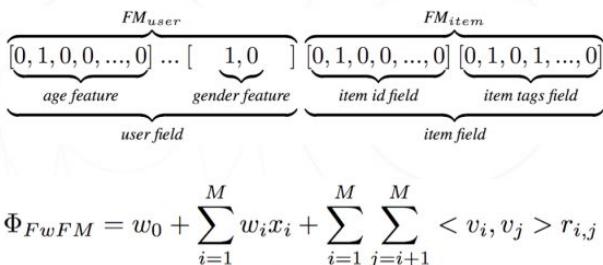
## 2.4 FM 模块

但是，MF 分解也会存在不足，例如：对于用户侧存在的丰富多样的特征，没有办法使用矩阵分解进行两两二阶交叉。

因此，在 field 内部引入 FM，用来显式学习用户侧 feature 粒度的所有特征的二阶交叉组合。具体来讲，给 user field 引入一个 FM 模型，对用户的年龄、性别等特征的二阶交叉，同样的算法也用于 item field 等。这样，FwFM 模型就演化成了下图这样一个 Field-wise Bi-Interaction 组件。引入 FM 模型后，模型的 AUC 指标提升了约 0.002。

## 03 排序——NFwFM(3/5)

- Field内引入FM, 显式学习二阶特征交叉
- AUC等离线指标提升 +0.002



$$\Phi_{FwBI} = w_0 + \underbrace{\sum_{i=1}^M w_i}_S + \underbrace{\sum_{i=1}^M \sum_{j=i+1}^M \left[ \left( \sum_{n, F(n)=i} \mathbf{v}_i \right) \left( \sum_{n, F(n)=j} \mathbf{v}_j \right) r_{i,j} \right]}_{FwFM} + \underbrace{\sum_{m=1}^M \left[ \left( \sum_{n, F(n)=m} \mathbf{v}_m \right)^2 - \left( \sum_{n, F(n)=m} (\mathbf{v}_m)^2 \right) \right] r_{m,m}}_{FM}$$

Field-wise Bi-Interaction

## 2.5 解决特征间干扰问题

但是这样并没有解决最一开始提到的问题：FM 模型在学习过程中，特征存在互相干扰的情况。

回顾一下前文所述的特征间干扰问题，即 FM 的 co-training 问题：

$$\nabla_{v_{\text{gender}}} f_{\text{FM}}(x) = v_{\text{age}} + v_{\text{item\_id}} + v_{\text{item\_tags}} + v_{\text{os}} + v_{\text{network}}$$

FM无法建模  $v_{\text{gender}} \perp v_{\text{network}}$

模型在对每一个特征进行学习的时候，都需要和其他特征进行交叉。例如，用户性别特征和网络环境特征应该是不相关的，但是模型在学习性别特征的时候不可避免地受到网络环境的影响。

为解决这一问题，借鉴 dropout 思路：模型训练完成 Bi-Interaction 后，按照伯努利分布（期望为  $\beta$ ）随机丢弃部分二阶交叉项，以解决部分 co-training 问题。

$$\Phi_{FM}_{\text{train}} = w_0 + \sum_{i=1}^N w_i + \sum_{i=1}^N \sum_{j=i+1}^N p \mathbf{v}_i \odot \mathbf{v}_j$$

$$\Phi_{FM}_{\text{inference}} = w_0 + \sum_{i=1}^N w_i + \sum_{i=1}^N \sum_{j=i+1}^N \beta \mathbf{v}_i \odot \mathbf{v}_j$$

where  $\mathbf{p} \in \mathcal{R}^{K_e}$  and  $p[i] \sim Bernoulli(\beta)$



具体来讲，先从伯努利分布中采样出由{0,1}组成的向量，再用该向量和 FM 模型计算得到的表示二阶特征交叉组合的向量进行相乘，这样可以随机丢弃部分二阶交叉项。在预估的时候是将 FM 做了 Bi-Interaction 后得到的向量，乘以伯努利分布的期望  $\beta$ ，用来对齐计算过程中丢失的数据的大小。引入如上思路之后，AUC 提升约 0.001。

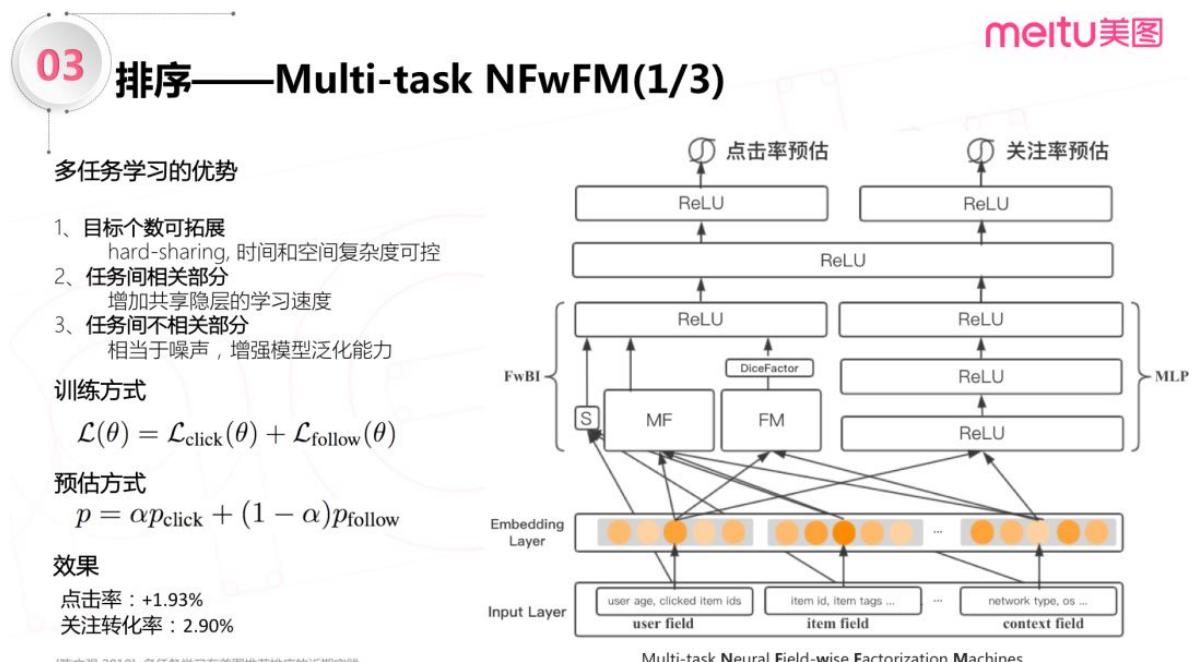
## 2.6 总结

总体来讲，我们从 FwFM 演化出 Field-wise Bi-Interaction 组件，包含线性加和模块用来学习一阶特征，还包括矩阵分解 (MF) 模块和 FM 模块，用来学习特征 field 粒度和 feature 粒度的特征交叉。相比于上一代 NFM 模型，在计算量和参数量都减少了 6 倍的情况下，点击率得到了 5.19% 的提升。

## 3. Multi-task NFwFM

### 3.1 多任务基本架构

接下来是多任务方面的工作。在深度学习时代，深度模型能够包含多种不同分布的样本，释放了多任务学习的最大价值。从实践角度考量，为使离线训练和在线预估效率较高，目标个数具有可拓展性。业界通常会选择下图中这种底层硬共享 (hard-sharing) 隐层的多任务架构，在这种架构下，因为点击率和关注转化率任务是强相关的工作，能增加共享隐层的学习速度，从而增加模型的收敛效率，而这两个任务中不相关的部分可以认为是相互任务的噪声，可以增强模型的泛化能力。



在学习的过程中，是利用两个任务简单加和的方式来学习多目标的。这个方式的离线 AUC 和单独的点击用户模型的 AUC 基本持平；线上点击率提升 1.93%，关注转化率提升 2.90%。

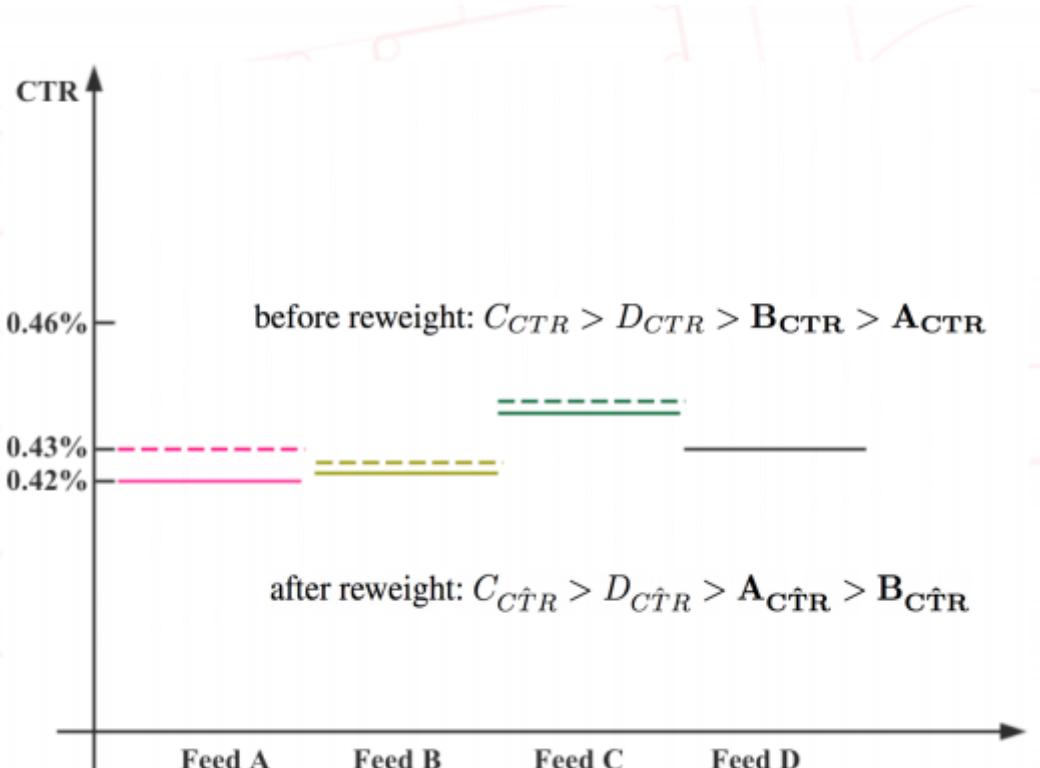
$$\mathcal{L}(\theta) = \mathcal{L}_{\text{click}}(\theta) + \mathcal{L}_{\text{follow}}(\theta)$$

在实际情况中，点击和关注的样本比例大概为100:1；在这样很少的关注样本的情况下，使用上述的多任务架构就可以得到稳定的提升；这驱使我们引入更多的关注数据，来压榨多任务模型学习更多更高质量数据的能力。

### 3.2 样本 reweight

具体来说，我们引入一个样本 reweight 的概念，主要目的是为了引入更多更高质量的关注行为数据。因为无数的经验告诉我们这是非常有效的做法。

以下图为例，假设有A、B、C、D这4个 item，图中实线部分表示 item 的真实 CTR (由大到小分别是 C > D > B > A)；而实际的关注转化率的关系是 A > C > B > D = 0。



我们用实际的关注转化率取 reweight 这个样本之后，得到下图虚线部分的结果：C > D > A > B，即原本点击率最低的 item A，在 reweight 之后由于关注转化率高而变得点击率比 B 更高，这样 item A 更容易被模型推荐出来，这样就能够提升整体的关注转化率，并且因为 C、D 等因为本身点击率较高或者没有关注行为，它们不受分布改变的影响，因此他们的 CTR 大小关系不受影响。换一种理解，我们是在仅仅改变了有关注 item 的点击率分布的基础上引入了更多更高质量的关注行为数据。对原来的点击率预估模型的侵入很少，整体点击率不会下跌太多。该工作提高关注转化率14.93%，但是点击率提升很少（约0.84%）。

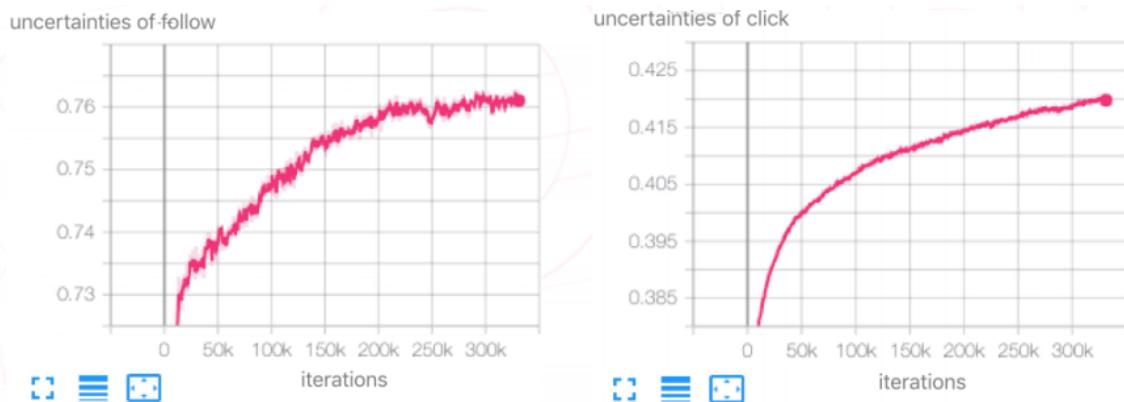
### 3.3 Homoscedastic Uncertainty 学习方式

上述工作点击率提升很少的原因是 reweight 模型过于简单。如前文所述，多任务的缺点是在参数共享的情况下，如果两个任务有不相关的部分，两个任务就会互相干扰，从而影响效果。学术界将这种现象称为共享冲突。共享冲突这一问题分析和解决起来较为复杂。针对美图的具体场景，减少共享冲突的一种方法是加大点击率预估任务的重要性，让点击率预估任务主导底部共享参数学习，进而让整体模型优先正确预估点击率模型，再去预估点击转化率任务。

在实践中，我们用同方差不确定性来学习每个任务对整体的主导能力。具体来说，分别给点击率任务和关注转化率任务各自一个参数 ( $\theta_{click}$  和  $\theta_{follow}$ ) 用来表示各自的不确定性；不确定性越小的任务对模型整体的主导性越强。

$$\mathcal{L}(\theta) = \frac{1}{2\sigma_{click}^2} \mathcal{L}_{click}(\theta) + \frac{1}{2\sigma_{follow}^2} \mathcal{L}_{follow}(\theta) + \log \sigma_{click}^2 \sigma_{follow}^2$$

从下图可以看出，左图的关注转化的不确定性达到0.76，确实比右图的点击率的不确定性（约0.42）更高；因此，让点击率预估任务主导整个模型的学习。



这样可以避免点击率下跌的风险（实际上，点击率提升了1.57%），而关注转化率的提升达到了15.65%。

#### 4. 总结

排序端的工作，美图经历了从 LR 模型到深度学习模型的引进：

- (1) 引入 NFM 模型，点击率提升了5.53%，人均时长提升6.97%
- (2) NFwFM 模型在引入了特征 Filed 信息后，在模型尺寸和计算复杂度可控的情况下，点击率提升了5.19%，人均时长提升了2.93%；

多目标 NFwFM 模型，在引入更多更高质量数据之后，不仅关注转化率提升了15.65%，点击率也提升了1.57%。

