# 0. 推荐算法常见知识

协同过滤(CF)、矩阵分解(MF)、Embedding+MLP、Wide&Deep、NeuralCF、DeepFM、GraphSAGE、DIN、DIEN、MIMN、DICM

**协同过滤**

**1．协同过滤的原理**

协同大家的反馈、评价和意见一起对海量的信息进行过滤，从中筛选出用户可能感兴趣的信息。在“协同”过滤算法中，推荐的原理是让用户考虑与自己兴趣相似用户的意见。

**2． 协同过滤的过程**

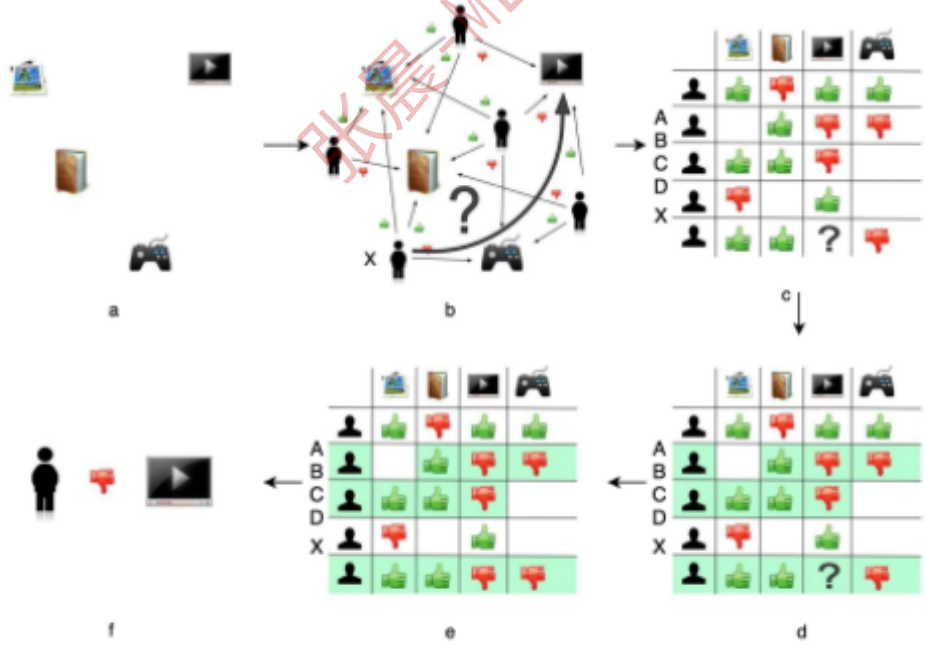



图1 协同过滤的过程 （来自《深度学习推荐系统》）

- 计算用户与商品的共现矩阵

- 计算用户之间的相似度，找出top-n个相似度大的用户
- 根据用户及用户对该商品的打分，计算将该商品推荐给用户x的概率

$$R_{u,p} = \frac{\sum_{s \epsilon S}(w_{u,s} \cdot R_{s,p})}{\sum_{s \in S} w_{u,s}}$$

其中，权重 $w_{u,s}$ 是用户 $u$ 和用户 $s$ 的相似度，$R_{s,p}$ 是用户 $s$ 对物品 $p$ 的评分。

### 3．协同过滤的缺点

虽然说协同过滤是目前公认的最经典的推荐算法，但我们还是可以轻松找出它的缺点，那就是共现矩阵往往非常稀疏，在用户历史行为很少的情况下，寻找相似用户的过程并不准确。

## 矩阵分解算法

### 1．矩阵分解原理

著名的视频流媒体公司 Netflix 对协同过滤算法进行了改进，提出了矩阵分解算法，加强了模型处理稀疏矩阵的能力。矩阵分解算法则是期望为每一个用户和视频生成一个隐向量，将用户和视频定位到隐向量的表示空间上（即用户向量与商品向量在同一空间中），距离相近的用户和视频表明兴趣特点接近，在推荐过程中，我们就应该把距离相近的视频推荐给目标用户。

### 2．矩阵分解算法的过程

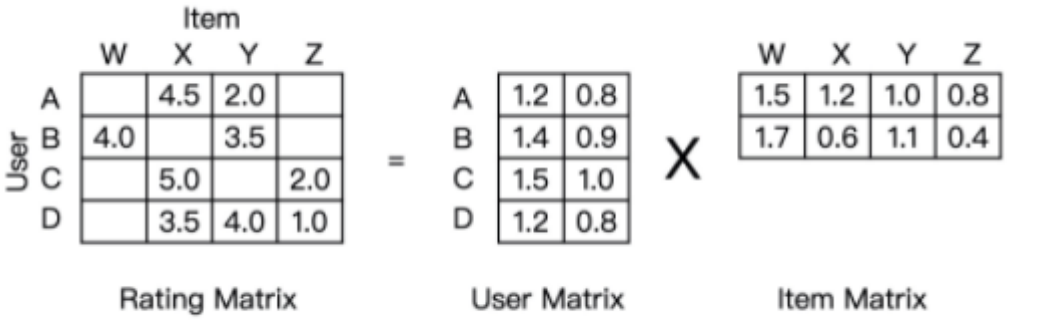



图3 矩阵分解示意图

- 计算用户与商品的共现矩阵
- 将矩阵分解，生成用户和物品的隐向量

- 通过用户和物品隐向量间的相似性进行推荐

**3．共现矩阵分解方法**

最常用的方法就是梯度下降。损失函数如下：

$$\min_{q^*,p^*} \sum_{(u,i)\in \mathrm{K}} \left(r_{ui} - q_i^{\mathrm{T}} p_u\right)^2$$

这个目标函数里面，rui 是共现矩阵里面用户 u 对物品 i 的评分，qi 是物品向量，pu 是用户向量，K 是所有用户评分物品的全体集合。简单来说就是，我们希望用户矩阵和物品矩阵的乘积尽量接近原来的共现矩阵。

**Embedding+MLP**

**1． Deep Crossing**

微软在 2016 年提出的深度学习模型 Deep Crossing，微软把它用于广告推荐这个业务场景上。它是一个经典的 Embedding+MLP 模型结构。

**2．Deep Crossing的结构**



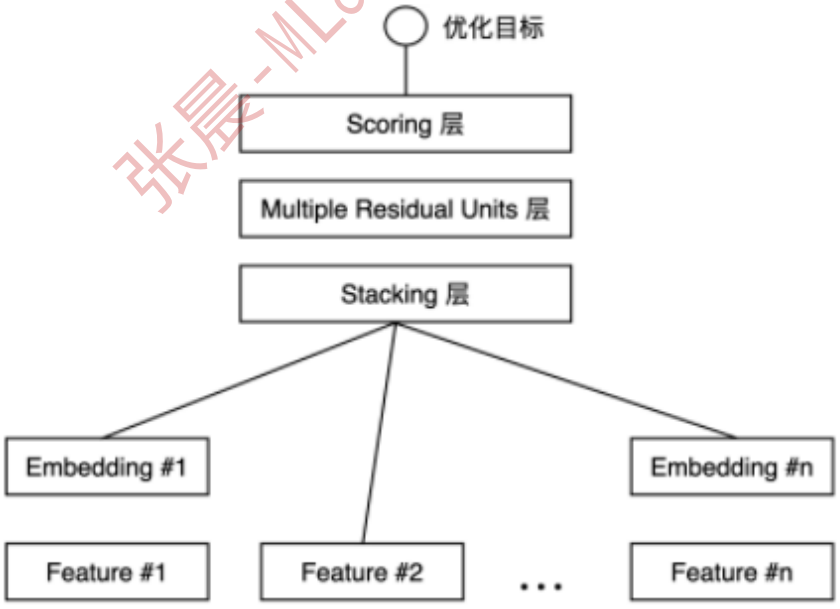



图1 经典的Embedding+MLP模型结构（图片来自 Deep Crossing - Web-Scale Modeling without Manually Crafted Combinatorial Features）

- Feature#1 代表的是类别型特征经过 One-hot 编码后生成的特征向量，而 Feature#2 代表的是数值型特征。
- One-hot 特征太稀疏了，不适合直接输入到后续的神经网络中进行训练，

所以我们需要通过连接到 Embedding 层的方式，把这个稀疏的 One-hot 向量转换成比较稠密的 Embedding 向量。

- Stacking 层中文名是堆叠层，我们也经常叫它连接（Concatenate）层。它的作用比较简单，就是把不同的 Embedding 特征和数值型特征拼接在一起，形成新的包含全部特征的特征向量。
- MLP 层就是我们开头提到的多层神经网络层，在图 1 中指的是 Multiple Residual Units 层，中文叫多层残差网络。微软在实现 Deep Crossing 时针对特定的问题选择了残差神经元。MLP 层的作用是让特征向量不同维度之间做充分的交叉，让模型能够抓取到更多的非线性特征和组合特征的信息。
- 最后是 Scoring 层，它也被称为输出层。可以采用逻辑回归作为输出层神经元。



**Wide&Deep**

**1．Wide&Deep模型结构**

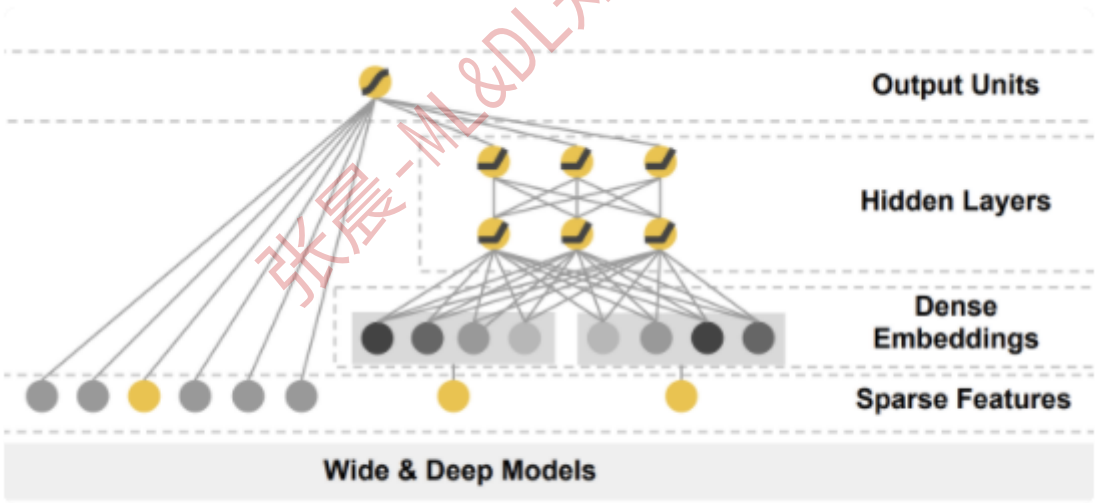



图1 Wide&Deep模型结构
(出自Wide & Deep Learning for Recommender Systems )

由左侧的 Wide 部分和右侧的 Deep 部分组成的。Wide 部分的结构太简单了，就是把输入层直接连接到输出层，中间没有做任何处理。Deep 层的结构稍复杂，但是本质就是 Embedding+MLP 模型结构。

**2．Wide&Deep各部分的作用**

Wide 部分的主要作用是让模型具有较强的“记忆能力”（Memorization），而 Deep 部分的主要作用是让模型具有“泛化能力”（Generalization），因

为只有这样的结构特点，才能让模型兼具逻辑回归和深度神经网络的优点，也就是既能快速处理和记忆大量历史行为特征，又具有强大的表达能力。

**3．为什么模型要有 Wide 部分**

因为 Wide 部分可以增强模型的记忆能力，让模型记住大量的直接且重要的规则，这正是单层的线性模型所擅长的。

所谓的 “记忆能力”，可以被宽泛地理解为模型直接学习历史数据中物品或者特征的“共现频率”，并且把它们直接作为推荐依据的能力 。就像我们在电影推荐中可以发现一系列的规则，比如，看了 A 电影的用户经常喜欢看电影 B，这种“因为 A 所以 B”式的规则，非常直接也非常有价值。但这类规则有两个特点：一是数量非常多，一个“记性不好”的推荐模型很难把它们都记住；二是没办法推而广之，因为这类规则非常具体，没办法或者说也没必要跟其他特征做进一步的组合。就像看了电影 A 的用户 80% 都喜欢看电影 B，这个特征已经非常强了，我们就没必要把它跟其他特征再组合在一起。

**4．为什么模型要有Deep部分**

“泛化能力”指的是模型对于新鲜样本、以及从未出现过的特征组合的预测能力。深度学习模型有很强的数据拟合能力，在多层神经网络之中，特征可以得到充分的交叉，让模型学习到新的知识。



**NeuralCF**

**1．NeuralCF的模型结构**

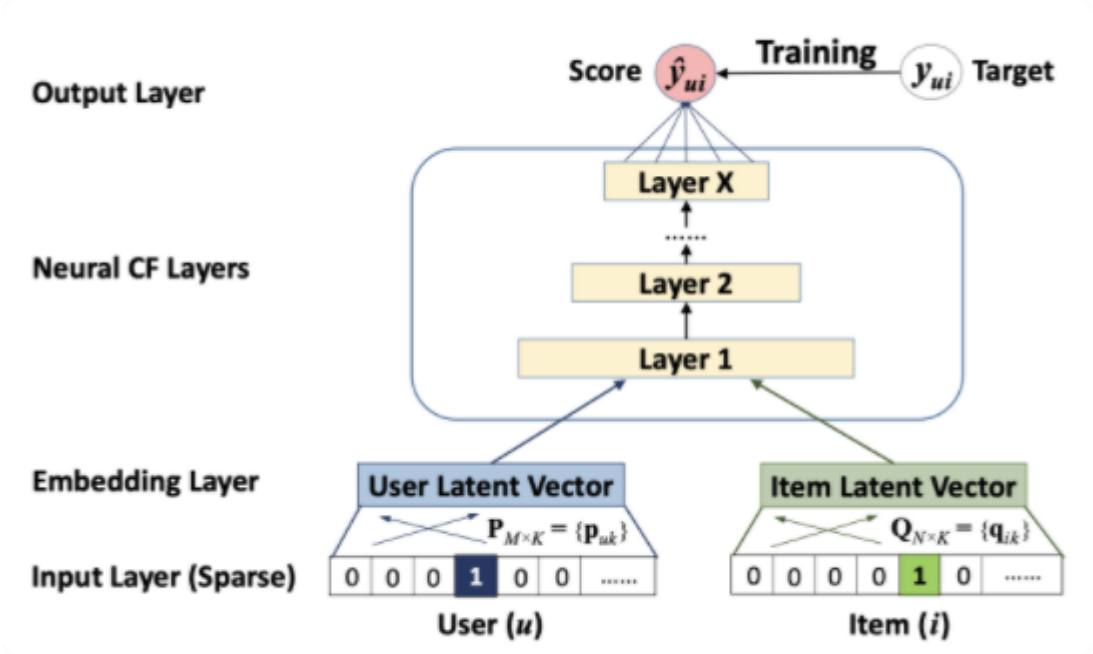



图3 NeuralCF的模型结构图 （出自论文Neural Collaborative Filtering）

Layer1是将用户emb与商品emb进行连接

LayerX是计算一个sigmoid得分，即计算出为用户推荐商品的概率

**2．NeuralCF与CF相比有什么优势**

协同过滤是通过直接利用非常稀疏的共现矩阵进行预测的，所以模型的泛化能力非常弱，遇到历史行为非常少的用户，就没法产生准确的推荐结果了。虽然，我们可以通过矩阵分解算法增强它的泛化能力，但因为矩阵分解是利用非常简单的内积方式来处理用户向量和物品向量的交叉问题的，所以，它的拟合能力也比较弱。使用深度学习网络来改进了传统的协同过滤算法，取名NeuralCF（神经网络协同过滤）。NeuralCF 大大提高了协同过滤算法的泛化能力和拟合能力。

**3．NeuralCF 模型的扩展，双塔模型**

NeuralCF 的模型结构之中，蕴含了一个非常有价值的思想，就是我们可以把模型分成用户侧模型和物品侧模型两部分，然后用互操作层把这两部分联合起来，产生最后的预测得分。这里的用户侧模型结构和物品侧模型结构，可以是简单的 Embedding 层，也可以是复杂的神经网络结构，最后的互操作层可以是简单的点积操作，也可以是比较复杂的 MLP 结构。但只要是这种物品侧模型 + 用户侧模型 + 互操作层的模型结构，我们把它统称为“双塔模型”结构。

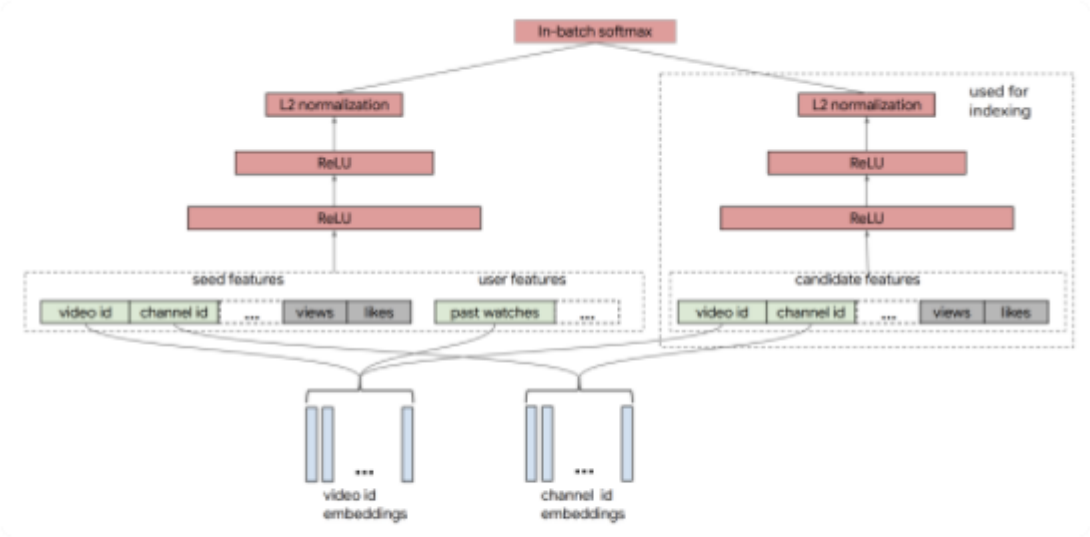



图5 YouTube双塔召回模型的架构
(出自论文 Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations)

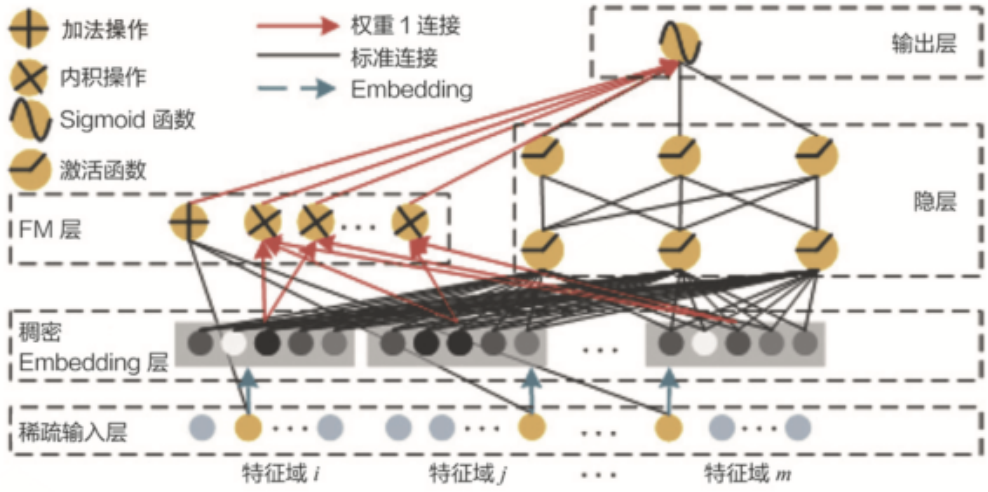
### 4．双塔模型相比我们之前学过的 Embedding MLP 和 Wide&Deep 有什么优势

在实际工作中，双塔模型最重要的优势就在于它易上线、易服务。物品塔和用户塔最顶端的那层神经元，那层神经元的输出其实就是一个全新的物品 Embedding 和用户 Embedding。我们就可以把 u(x) 和 v(y) 存入特征数据库，这样一来，线上服务的时候，我们只要把 u(x) 和 v(y) 取出来，再对它们做简单的互操作层运算就可以得出最后的模型预估结果了。所以使用双塔模型，我们不用把整个模型都部署上线，只需要预存物品塔和用户塔的输出，以及在线上实现互操作层就可以了。



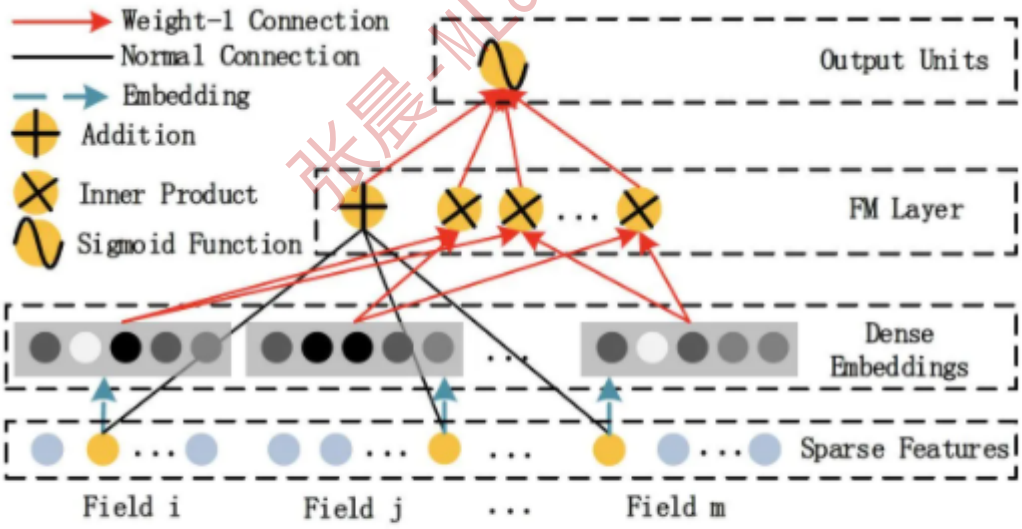
## DeepFM

### 1．DeepFM的模型结构




这是一个全新的既有强特征组合能力，又有强拟合能力的模型。DeepFM 利用了 Wide&Deep 组合模型的思想，用 FM 替换了 Wide&Deep 左边的 Wide 部分，加强了浅层网络部分特征组合的能力，而右边的部分跟 Wide&Deep 的 Deep 部分一样，主要利用多层神经网络进行所有特征的深层处理，最后的输出层是把 FM 部分的输出和 Deep 部分的输出综合起来，产生最后的预估结果。这就是 DeepFM 的结构。

**2．FM结构**




The architecture of FM.

图1 FM的神经网络化结构

(出自论文 DeepFM: A Factorization-Machine based Neural Network for CTR Prediction)

FM 会使用一个独特的层 FM Layer 来专门处理特征之间的交叉问题。FM 层中有多个内积操作单元对不同特征向量进行两两组合，这些操作单元会把不同特

征的内积操作的结果输入最后的输出神经元，以此来完成最后的预测。

### 3．FM结构的优势

特征交叉能力强。无论是 Embedding MLP，还是 Wide&Deep 其实都没有对特征交叉进行特别的处理，而是直接把独立的特征扔进神经网络，让它们在网络里面进行自由组合。

### 4．为什么深度学习模型需要加强处理特征交叉的能力

MLP 有拟合任意函数的能力，但这是建立在 MLP 有任意多层网络，以及任意多个神经元的前提下的。在训练资源有限，调参时间有限的现实情况下，MLP 对于特征交叉的处理其实还比较低效。因为 MLP 是通过 concatenate 层把所有特征连接在一起成为一个特征向量的，这里面没有特征交叉，两两特征之间没有发生任何关系。

## DIN

### 1．BASE模型的结构



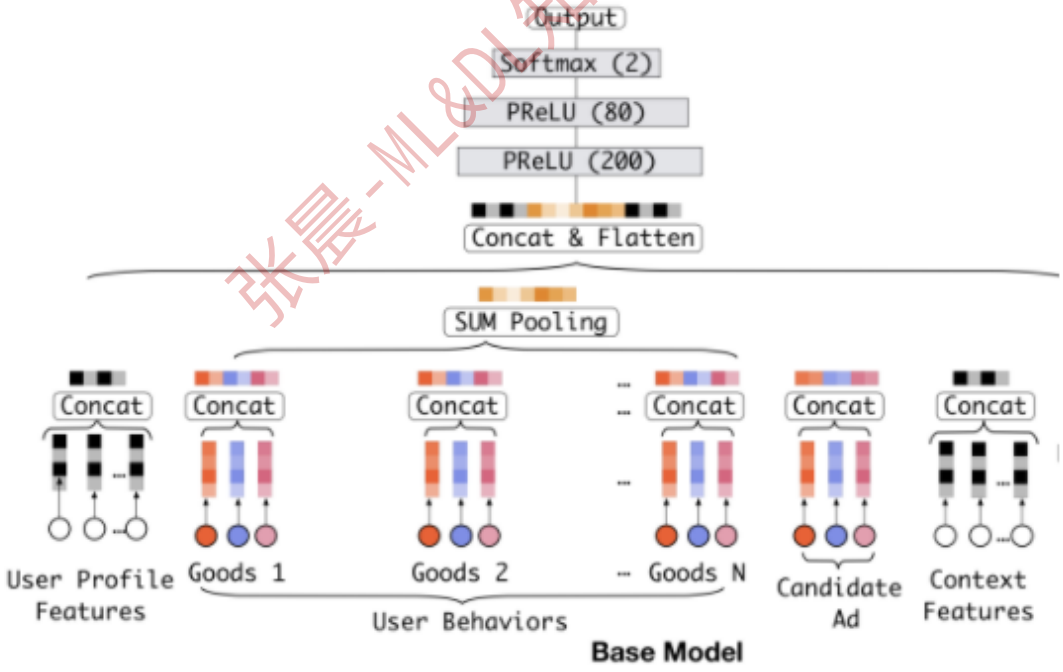



图2 阿里Base模型的架构图
(出自论文 Deep Interest Network for Click-Through Rate Prediction)

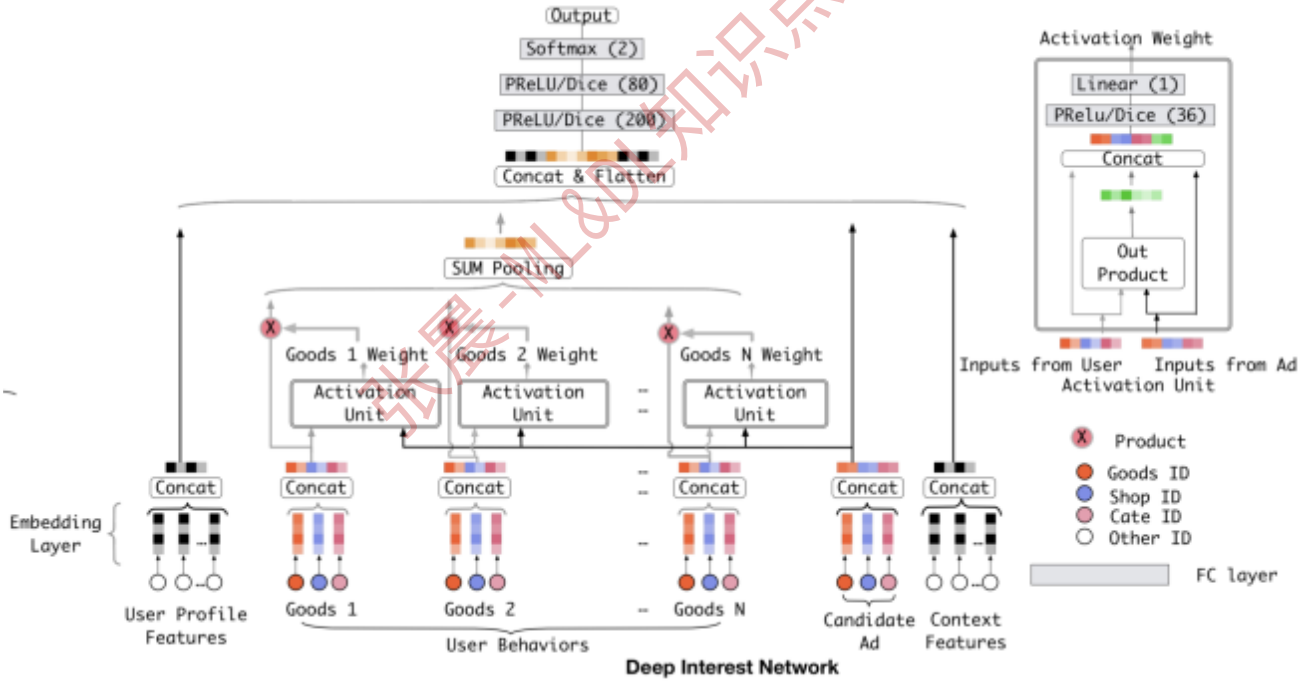
Base Model 是一个典型的 Embedding MLP 的结构。它的输入特征有用户属性特征（User Proflie Features）、用户行为特征（User Behaviors）、候选广告特征（Candidate Ad）和场景特征（Context Features）。用户行为特征是

由一系列用户购买过的商品组成的，也就是图上的 Goods 1 到 Goods N，而每个商品又包含了三个子特征，也就是图中的三个彩色点，其中红色代表商品 ID，蓝色是商铺 ID，粉色是商品类别 ID。同时，候选广告特征也包含了这三个 ID 型的子特征，因为这里的候选广告也是一个阿里平台上的商品。

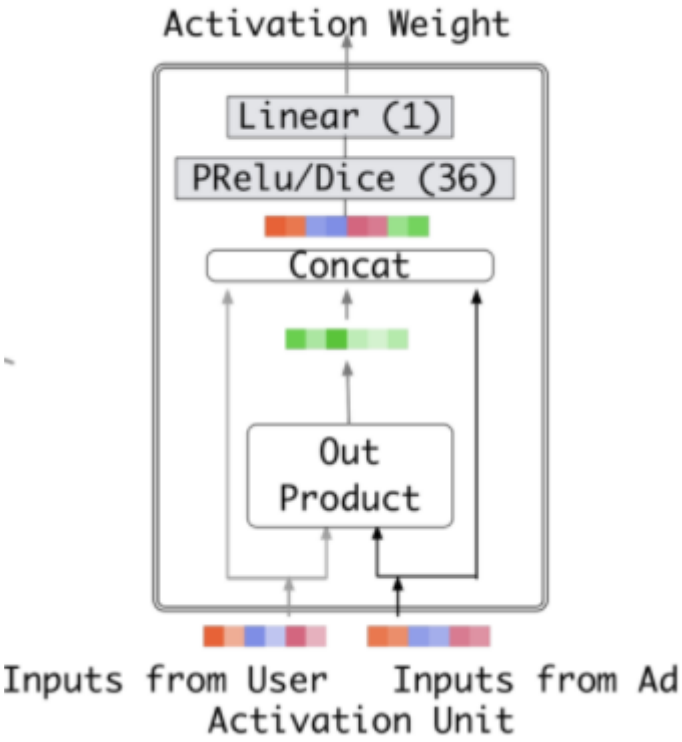
因为用户的行为序列其实是一组商品的序列，这个序列可长可短，但是神经网络的输入向量的维度必须是固定的，SUM Pooling 层的结构就是直接把这些商品的 Embedding 叠加起来，然后再把叠加后的 Embedding 跟其他所有特征的连接结果输入 MLP。

### 2．DIN模型结构

SUM Pooling 的 Embedding 叠加操作其实是把所有历史行为一视同仁，没有任何重点地加起来，这其实并不符合我们购物的习惯。阿里正是在 Base Model 的基础上，把注意力机制应用在了用户的历史行为序列的处理上，从而形成了 DIN 模型。




### 3．如何使用注意力机制的

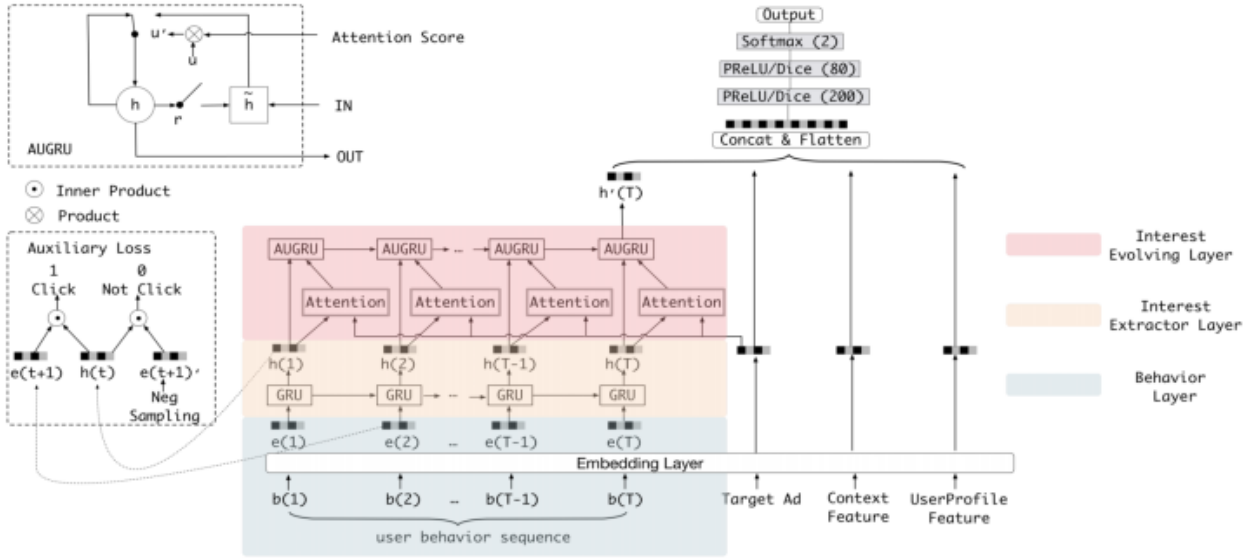



它的输入是当前这个历史行为商品的 Embedding，以及候选广告商品的 Embedding。我们把这两个输入 Embedding，与它们的外积结果连接起来形成一个向量，再输入给激活单元的 MLP 层，最终会生成一个注意力权重，这就是激活单元的结构。简单来说，激活单元就相当于一个小的深度学习模型，它利用两个商品的 Embedding，生成了代表它们关联程度的注意力权重。



## DIEN

### 1．DIEN的结构



- 最下面一层是行为序列层（Behavior Layer，浅绿色部分）。它的主要作用和一个普通的 Embedding 层是一样的，负责把原始的 ID 类行为序列转换成 Embedding 行为序列。
- 再上一层是兴趣抽取层（Interest Extractor Layer，浅黄色部分）。它的主要作用是利用 GRU 组成的序列模型，来模拟用户兴趣迁移过程，抽取出每个商品节点对应的用户兴趣。
- 最上面一层是兴趣进化层（Interest Evolving Layer，浅红色部分）。它的主要作用是利用 AUGRU(GRU with Attention Update Gate) 组成的序列模型，在兴趣抽取层基础上加入注意力机制，模拟与当前目标广告（Target Ad）相关的兴趣进化过程，兴趣进化层的最后一个状态的输出就是用户当前的兴趣向量 h'(T)。

**2．DIEN对DIN的改进**

无论是电商购买行为，还是视频网站的观看行为，或是新闻应用的阅读行为，特定用户的历史行为都是一个随时间排序的序列。既然是和时间相关的序列，就一定存在前后行为的依赖关系，这样的序列信息对于推荐过程是非常有价值的。DIEN 模型正好弥补了 DIN 模型没有对行为序列进行建模的缺陷，它围绕兴趣进化这个点进一步对 DIN 模型做了改进。