[论文阅读]阿里DIN深度兴趣网络之总体解读

原创 罗西的思考 罗西的思考 10月17日

[论文阅读]阿里DIN深度兴趣网络之总体解读

0x00 摘要

Deep Interest Network（DIN）是阿里妈妈精准定向检索及基础算法团队在2017年6月提出的。其针对电子商务领域（e-commerce industry）的CTR预估，重点在于充分利用/挖掘用户历史行为数据中的信息。

本系列文章会解读论文以及源码，顺便梳理一些深度学习相关概念和TensorFlow的实现。

本文是系列第一篇： 论文解读。参考了大量文章，衷心感谢各位的分享，具体请参见文末链接。

0x01 论文概要

1.1 概括

Deep Interest Network（DIN）是阿里妈妈精准定向检索及基础算法团队在2017年6月提出的。其针对电子商务领域（e-commerce industry）的CTR预估，重点在于充分利用/挖掘用户历史行为数据中的信息。

DIN通过引入attention机制，针对不同的广告构造不同的用户抽象表示，从而实现了在数据维度一定的情况下，更精准地捕捉用户当前的兴趣。

核心思想是： 用户的兴趣是多元化的（**diversity**），并且对于特定的广告，用户不同的兴趣会产生不同的影响（**local activation**）。

1.2 文章信息

- 论文标题: Deep Interest Network for Click-Through Rate Prediction

- 论文地址: https://arxiv.org/abs/1706.06978
- 代码地址: https://github.com/zhougr1993/DeepInterestNetwork，另外作者在 README 中推荐看 https://github.com/mouna99/dien 中的实现

1.3 核心观点

文章介绍了现有的点击率 （CTR） 预估模型大都满足相同的模式:

- 先将大量的稀疏类别特征 （Categorical Features） 通过 Embedding 技术映射到低维空间;
- 再将这些特征的低维表达按照特征的类别进行组合与变换 （文中采用 in a group-wise manner 来描述），以形成固定长度的向量（比如常用的 sum pooling / mean pooling）；
- 最后将这些向量 concatenate 起来输入到一个 MLP （Multi-Layer Perceptron）中，从而学习这些特征间的非线性关系;

这个模式存在一个问题。比如在电商场景下，用户兴趣可以使用用户的历史行为来描述 （比如用户访问过的商品，店铺或者类目），然而如果按照现有的处理模式，对于不同的候选广告，用户的兴趣始终被映射为同一个固定长度的向量来表示，这极大的限制了模型的表达能力，毕竟用户的兴趣是多样的。

Embedding&MLP模型的瓶颈就是表达用户多样的兴趣，维度受限的用户表示向量将成为表达用户多样化兴趣的瓶颈。

为了解决这个问题，论文中提出了 DIN 网络。对于不同的候选广告，考虑该广告和用户历史行为的相关性，以便自适应地学习用户兴趣的特征表达。具体来说，文章介绍了 local activation unit 模块，其基于 Attention 机制，对用户历史行为进行加权来表示用户兴趣，其中权重参数是通过候选广告和历史行为交互来进行学习的。

另外，本文还介绍了 Mini-batch Aware Regularization 与 Dice 激活函数两种技术，以帮助训练大型的网络。

1.4 名词解释

Diversity：
用户在访问电商网站时会对多种商品都感兴趣。也就是用户的兴趣非常的广泛。比如一个年轻的母亲，从她的历史行为中，我们可以看到她的兴趣非常广泛：羊毛衫、手提袋、耳环、童装、运动装等等。

Local Activation：

用户是否会点击推荐给他的商品，仅仅取决于历史行为数据中的一小部分，而不是全部。历史行为中部分数据主导是否会点击候选广告。比如一个爱游泳的人，他之前购买过travel book、ice cream、potato chips、swimming cap。当前给他推荐的商品（或者说是广告Ad）是goggle（护目镜）。那么他是否会点击这次广告，跟他之前是否购买过薯片、书籍、冰激凌一丁点关系也没有！而是与他之前购买过游泳帽有关系。也就是说在这一次CTR预估中，部分历史数据（swimming cap）起了决定作用，而其他的基本没啥用。

0x02 解读思路

下面主要摘录：用NumPy手工打造 Wide & Deep。

2.1 Memorization 和 Generalization

推荐系统的主要挑战之一，是同时解决Memorization和Generalization。Memorization根据历史行为数据，推荐通常和用户已有行为的物品直接相关的物品。而Generalization会学习新的特征组合，提高推荐物品的多样性。DeepFM 中 Wide & Deep 分别对应 Memorization & Generalization。

2.1.1 Memorization

面对拥有大规模离散sparse特征的CTR预估问题时，将特征进行非线性转换，然后再使用线性模型是在业界非常普遍的做法，最流行的即「LR+特征叉乘」。Memorization 通过一系列**人工的特征叉乘（cross-product）**来构造这些非线性特征，捕捉sparse特征之间的高阶相关性，即 “记忆” 历史数据中曾共同出现过的特征对。

例如

```
特征1—专业：{计算机、人文、其他}，

特征2—下载过音乐《消愁》:{是、否}，
```

这两个特征one-hot后的特征维度分别为3维与2维，对应的叉乘结果是

```
特征3—专业X下载过音乐《消愁》：{计算机∧是，计算机∧否，人文∧是，人文∧否，其他∧是，其他∧否
```

典型代表是LR模型，使用大量的原始sparse特征和**叉乘特征**作为输入，很多原始的dense特征通常也会被分桶离散化构造为sparse特征。这种做法的优点是模型可解释高，实现快速高效，特征重要度易于分析，在工业界已被证明是很有效的。

Memorization的缺点是：

- 需要更多的人工设计；
- 可能出现过拟合。可以这样理解：如果将所有特征叉乘起来，那么几乎相当于纯粹记住每个训练样本，这个极端情况是最细粒度的叉乘，我们可以通过构造更粗粒度的特征叉乘来增强泛化性；
- 无法捕捉训练数据中未曾出现过的特征对。例如上面的例子中，如果每个专业的人都没有下载过《消愁》，那么这两个特征共同出现的频次是0，模型训练后的对应权重也将是0；

2.1.2 Generalization

Generalization 为sparse特征学习低维的dense embeddings 来捕获特征相关性，学习到的embeddings 本身带有一定的语义信息。可以联想到NLP中的词向量，不同词的词向量有相关性，因此Generalization是基于相关性之间的传递。这类模型的代表是DNN和FM。

Generalization的优点是更少的人工参与，对历史上没有出现的特征组合有更好的泛化性 。

在推荐系统中，当user-item matrix非常稀疏时，例如有和独特爱好的users以及很小众的items，NN很难为users和items学习到有效的embedding。这种情况下，大部分user-item应该是没有关联的，但dense embedding 的方法还是可以得到对所有 user-item pair 的非零预测，因此导致*over-generalize*并推荐不怎么相关的物品。此时Memorization就展示了优势，它可以“记住”这些特殊的特征组合。

2.2 发展脉络

各种NN与FM看似繁杂，实际上，只要把握住它们的发展脉络，即**“如何兼顾记忆与扩展”、“如何处理高维、稀疏的类别特征”、“如何实现特征交叉”**，你就会发现各种高大上的新算法不过是沿着这条脉络，在某个枝叉上的修补。这样一来，各种NN与FM，在你脑中，就不再是一个个独立的缩写，而能够编织成网，融会贯通。

相比于实数型特征，**稀疏的类别/ID类特征，才是推荐、搜索领域的“一等公民”**，被研究得更多。即使有一些实数值特征，比如历史曝光次数、点击次数、CTR之类的，也往往通过bucket的方式，变成categorical特征，才喂进模型。

但是，稀疏的categorical/ID类特征，也有着**单个特征表达能力弱、特征组合爆炸、分布不均匀导致受训程度不均匀**的缺点。为此，一系列的新技术被开发出来。

单个categorical/ID特征表达能力是极弱的，因此必须做特征交叉，以增强categorical特征的表达能力。而围绕着如何做特征交叉，衍生出各种算法。

深度神经网络（DNN）先将categorical/id特征通过embedding映射成稠密向量，再喂入DNN，让DNN自动学习到这些特征之间的**深层交叉**，以增强扩展能力。

0x03 DNN

3.1 深度模型思路

准确的CTR预估需要精细化权衡用户、广告主、平台三方利益。经过多年的技术更新迭代与发展，CTR预估技术经历了从 LR/FM 到 融合模型（RF/GBDT/XGBoost） 到 深度CTR预估模型（FNN/PNN/WDL/DeepFM/DIN）的过程，而贯穿其中的主线是**如何让模型自动地进行组合特征的挖掘?**

比如：

- Wide&Deep、DeepFM：采用高阶和低阶特征的联合来提高模型的表达能力；
- PNN：在MLP之前引入一个乘积层（内积和外积），强调了特征Embedding向量之间的交叉方式，让模型更容易捕获特征的交叉信息；

也可以看看阿里思路的出发点：

> 我们第一考虑到的是降维，在降维的基础上，进一步考虑特征的组合。所以DNN很自然进入了我们的考虑范围。再考虑的是如果把用户行为序列建模起来，我们希望是用户打开手淘后，先在有好货点了一个商品，再在猜你希望点了一个商品，最后进入搜索后会受到之前的行为的影响，当然有很多类似的方法可以间接实现这样的想法。但直接建模的话，LR这类的模型，很难有能力来支持这类特征，所以很容易就想到了RNN模型。

3.2 DNN模型

DNN模型大多遵从 **Embedding + MLP**这一基础网络架构，即将原始高维的不同的离散特征映射为固定长度的低维embedding向量，并将embedding向量作为多个全连接层的输入，拟合高阶的

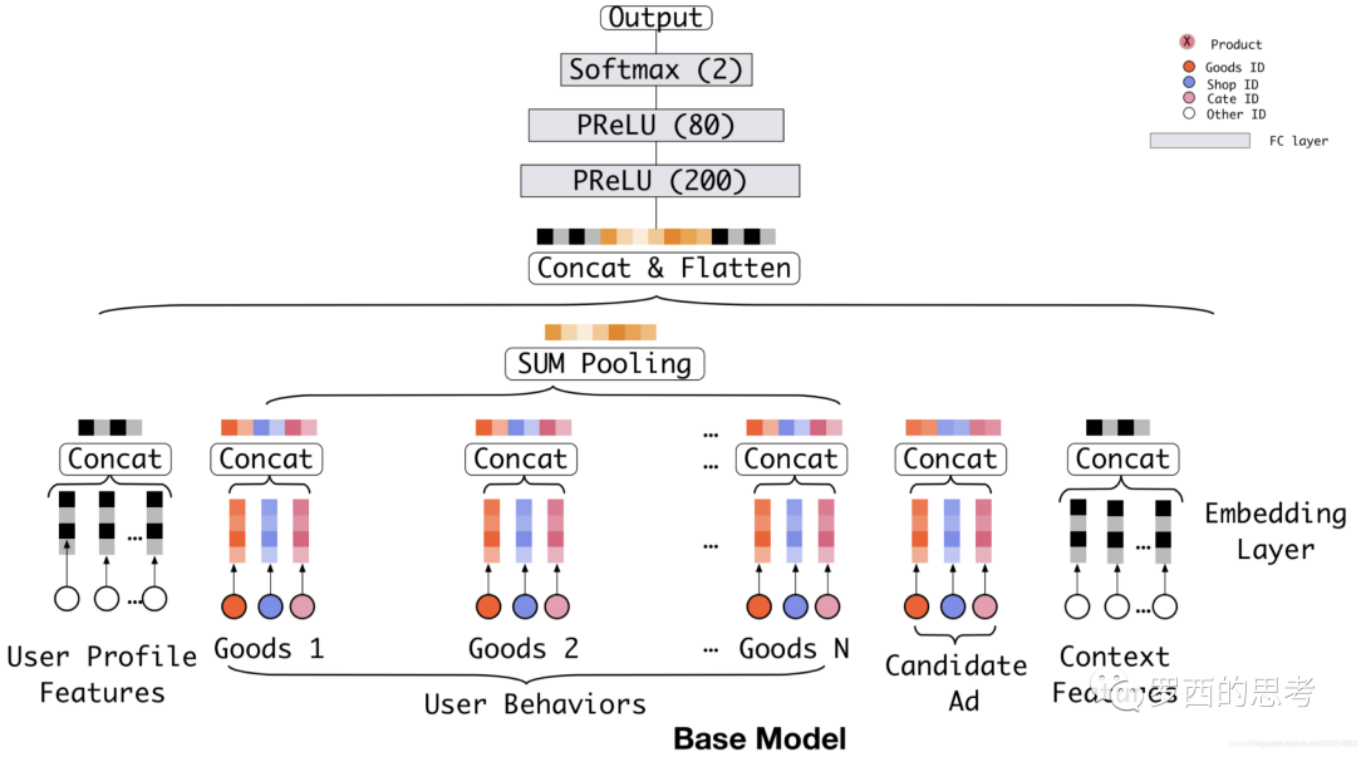
非线性关系，最后通过Sigmoid等手段将输出值归一到0~1，表示点击概率。相比于传统的LR、GBDT、FM等模型，这类DNN的模型能减少大量的人工构造特征过程，并且能学习特征之间的非线性关系。

通常流程是:

```
Sparse Features -> Embedding Vector -> pooling layer -> MLPs -> Sigmoid -> Output
```

3.3 工作机制

下图的 Base Model, 是现有的大多数 CTR 模型采用的模式:



在这里插入图片描述

红蓝粉三色节点分别表示商品 ID （Goods ID）, 店铺 ID （Shop ID）, 类目 ID （Cate ID） 三种稀疏特征, 其他的输入特征, 使用白色节点表示 （比如左边的用户特征, 比如用户 ID; 还有右边的上下文特征, 比如广告位之类的特征）. 注意 Goods 1 ~ Goods N 用来描述用户的历史行为. 候选广告 Candidate Ad 本身也是商品, 也具有 Goods / Shop / Cate ID 三种特征.

自底向上观测 Base Model 的工作机制：

- 第一模块：特征表示。
- 可将特征大致分为四类：user profile、user behavior、ad 以及 context 部分。

- 将广告设为目标。
- 每一类特征包含多个field，用户信息包含性别、年龄等等；用户行为包含用户访问过的物品编号；广告包含广告id，商店id等；上下文包含设计类型id，时间等等。
- 有的特征可以被编码成one-hot表示，例如女性可以被编码成[0,1]。有的特征可以进行 multi-hot 编码，与 one-hot 编码不同，multi-hot 编码中，一个向量可能存在多个 1。
- 在CTR序列模型中，值得注意的是每个字段都包含一个行为列表，每个行为对应一个one-hot向量。
- 第二模块：嵌入层。
- 学习特征的低维向量表示，将维数较大的稀疏特征矩阵转换成低维稠密特征矩阵。
- 每一个field都有一个独立的 embedding matrix。
- 值得注意的是，由于每个用户的历史行为数据各不相同，因此 e 的列数是不确定的。相应地也就不能直接与其他field的嵌入向量首尾相接 作为MLP层的输入。
- 第三模块：pooling层。
- 由于不同的用户有不同个数的行为数据，导致embedding矩阵的向量大小不一致，而全连接层只能处理固定维度的数据，因此利用Pooling Layer得到一个固定长度的向量。
- 本层对 e 进行sum pooling，即将一个类别的embedding向量输入进池化操作，转化为一个固定长度的向量，解决维度不定的问题。
- 第四模块：链接层。
- 经过embedding layer和pooling layer后，原始稀疏特征被转换成多个固定长度的用户兴趣的抽象表示向量。
- 然后利用concat layer聚合抽象表示向量，输出该用户兴趣的唯一抽象表示向量；作为 MLP 层的输入 。
- 第五模块：MLP 层，将concat layer输出的抽象表示向量作为MLP的输入，自动学习数据之间的高阶交叉特征。
- 损失函数 ： 基于深度学习的CTR模型广泛使用的损失函数是 负对数似然函数（the negative log-likelihood function）Loglos，使用标签作为目标项来监督整体的预测。

3.4 模型特点

优点:

- 通过神经网络可以拟合高阶的非线性关系, 同时减少了人工特征的工作量。

缺点:

- 表示用户的兴趣多样性有限制 （这是最大的瓶颈） 。在对用户历史行为数据进行处理时, 每个用户的历史点击个数是不相等的, 包含了许多兴趣信息，如何对用户多种多样的兴趣建模？我们要把它们编码成一个固定长的向量（这个向量就是用户表示，是用户兴趣的代表），需要做pooling （sum or average）, 会损失信息。比如：
- K维向量，最多只能表达K个独立的兴趣，而用户的兴趣可能不止K；
- K的大小会对计算量产生明显影响，一般用大的K效果会更好，即扩展向量的维度，但这样会增加学习的参数和在有限的数据中有过拟合的风险；
- 没有考虑用户与广告之间的关系。在电子商务领域中，用户的历史行为数据（User Behavior Data）中包含大量的用户兴趣信息，之前的研究并没有针对Behavior data**特殊的结构（Diversity + Local Activation）**进行建模。比如 对于同一个用户, 如果候选广告（Candidate Ad） 发生了变化, 用户的兴趣却依然是同一个向量来表达, 显然这限制了模型的表达能力, 毕竟用户的兴趣是丰富的/变化的。
- 忽略隐式特征的挖掘和表示。DNN模型直接将用户的行为视作用户的兴趣。行为是兴趣的载体，能反映兴趣，但若直接用行为表示兴趣则略有不妥。因为，行为是序列化产生的，如果像大部分现有的模型那样直接采用行为即兴趣的做法，会忽略行为之间的依赖关系。此外，当前时刻的兴趣往往直接导致了下一行为的发生。
- 忽略兴趣的变化。如之前所讲，用户的兴趣是不断变化的。例如用户对衣服的喜好，会随季节、时尚风潮以及个人品味的变化而变化，呈现一种连续的变迁趋势。但在淘宝平台中，用户的兴趣是丰富多样的，且每个兴趣的演变基本互不影响。此外，影响最终行为的仅仅是与目标商品相关的兴趣。
- 不必将某个用户所有的兴趣【用户的历史购买记录】全部压缩到向量中，因为只有用户部分的兴趣会影响当前行为（对候选广告点击或不点击）。例如，一位女游泳运动员会点击推荐的护目镜，这主要是由于购买了泳衣而不是上周购物清单中的鞋子。

0x04 DIN

针对DNN模型的问题，阿里提出了DIN模型。其核心思想：用户的兴趣是多元化的（**diversity**），并且对于特定的广告，用户不同的兴趣会产生不同的影响（**local activation**）。DIN同时对Diversity和Local Activation进行建模。

DIN 不会通过使用同一向量来表达所有用户的不同兴趣，而是通过考虑**历史行为的相关性**来自适应地计算用户兴趣的表示向量（对于给定的广告）。**该表示向量随不同广告而变化。**DIN 通过考虑【给定的候选广告】和【用户的历史行为】的相关性，来计算用户兴趣的表示向量。具体来说就是通过引入局部激活单元，通过软搜索历史行为的相关部分来关注相关的用户兴趣，并采用加权和来获得有关候选广告的用户兴趣的表示。与候选广告相关性较高的行为会获得较高的激活权重，并支配着用户兴趣。该表示向量在不同广告上有所不同，大大提高了模型的表达能力。

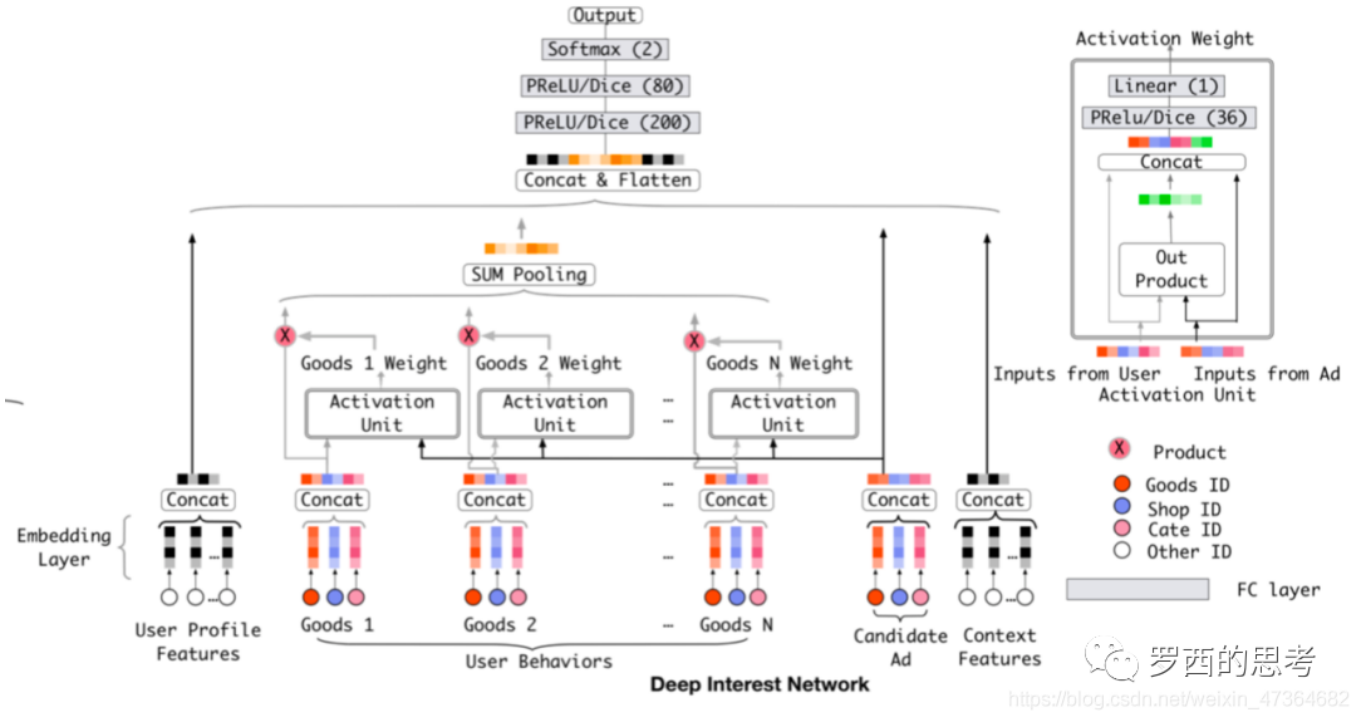
4.1 创新

Deep Interest NetWork有以下几点创新：

- **针对Diversity：** 针对用户广泛的兴趣，DIN用*an interest distribution*去表示，即用 Pooling（weighted sum）对Diversity建模（对用户多种多样的兴趣建模）。
- **针对Local Activation**：
- DNN 直接求sum或average损失了很多信息。所以 DIN 稍加改进，利用attention机制实现Local Activation，从用户历史行为中动态学习用户兴趣的embedding向量，针对不同的广告构造不同的用户抽象表示，从而实现了在数据维度一定的情况下，更精准地捕捉用户当前的兴趣。
- 对用户历史行为进行了不同的加权处理，针对不同的广告，不同的 behavior id 赋予不同的权重，这个权重是由当前behavior id和候选广告共同决定的，这就是Attention机制。即针对当前候选Ad，去局部的激活（*Local Activate*）相关的历史兴趣信息。
- 与当前候选Ad相关性越高的历史行为，会获得越高的*attention score*，从而会主导这一次预测。
- CTR中**特征稀疏而且维度高**，通常利用L1、L2、Dropout等手段防止过拟合。由于传统L2正则计算的是全部参数，CTR预估场景的模型参数往往数以亿计。DIN提出了一种正则化方法，在每次小批量迭代中，给与不同频次的特征不同的正则权重；
- 由于传统的**激活函数**，如Relu在输入小于0时输出为0，将导致许多网络节点的迭代速度变慢。PRelu虽然加快了迭代速度，但是其分割点默认为0，实际上分割点应该由数据决定。因此，DIN提出了一种数据动态自适应激活函数Dice。
- **针对大规模稀疏数据的模型训练：**当DNN深度比较深（参数非常多），输入又非常稀疏的时候，很容易过拟合。DIN提出**Adaptive regularizaion**来防止过拟合，效果显著。

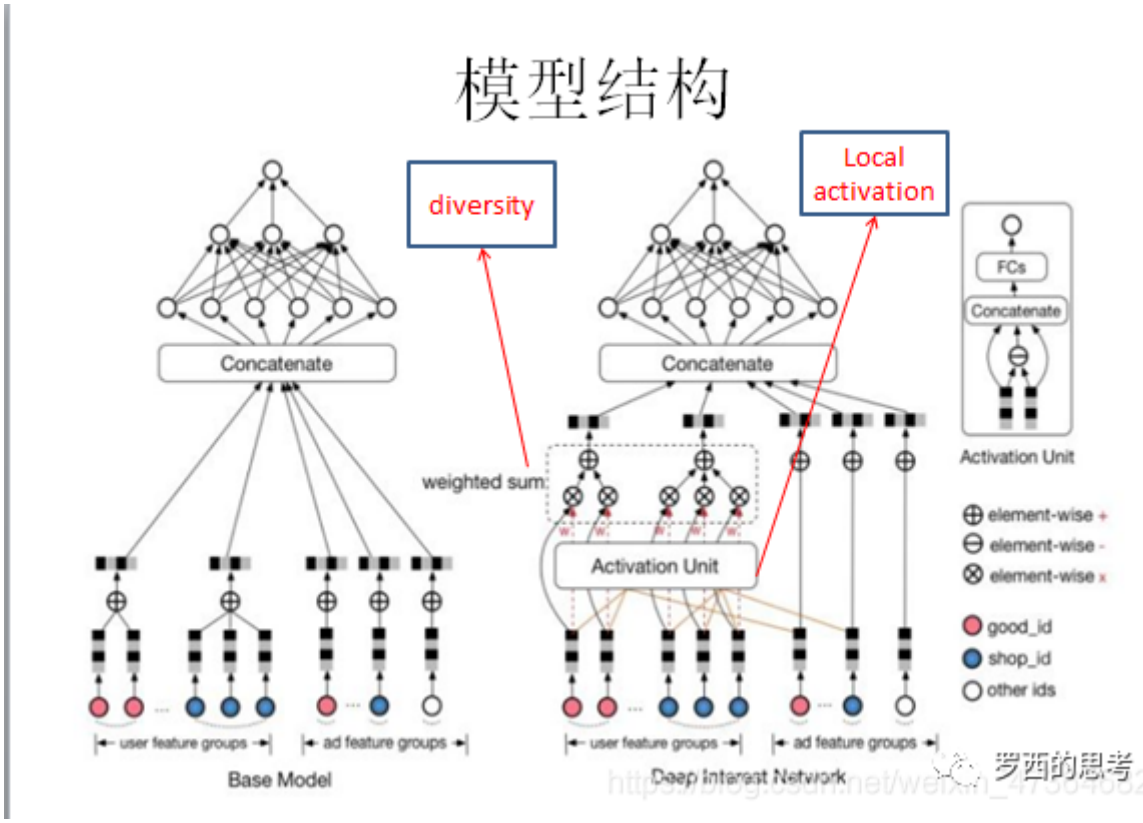
4.2 架构

DIN架构图如下：



在这里插入图片描述

DIN同时对Diversity和Local Activation进行建模，具体体现如下图。



在这里插入图片描述

下面我们逐一看看系统的各个部分。

0x05 特征

5.1 特征分类

论文中作者把阿里的展示广告系统特征分为四大类。

1）用户画像特征；

2）用户行为特征，即用户点击过的商品，各个用户行为长度不同；

3）待曝光的广告，广告其实也是商品；

4）上下文特征；

每个特征类别包括多个特征域（feature field），例如：用户画像特征包括性别，年龄段等；用户行为特征，包括用户点击过的商品，商品的类别，以及所属的商铺等；Context包括时间。

5.2 输入特点

CTR中输入普遍存在的特点：

- 高纬度
- 非常稀疏

有一些特征域是单值特征，不同的特征值之间是互斥的，例如性别只可能属于男或者女，可以转化为one-hot表示；

有一些特征域是多值离散特征，例如用户行为特征，用户可能点击过多个商品，构成一个商品点击序列，只能用multi-hot编码表示。与 one-hot 编码不同，multi-hot 编码中，一个向量可能存在多个 1，比如：

- 用户在YouTube上看的视频和搜索过的视频。无论是看过的还是搜索过的，都不止一个，但是相对于所有的视频来说，看过和搜索过的数量都太小了（非常稀疏）。
- 在电子商务上的例子就是：用户购买过的good_id有多个，购买过的shop_id也有多个，而这也直接导致了每个用户的历史行为id长度是不同的。

5.3 特征处理

DNN 并**没有进行特征组合/交叉特征**。而是通过DNN去学习特征间的交互信息。

对于单值特征处理比较简单，对于多值特征的处理稍微麻烦些。多值特征导致了每个用户的样本长度都是不同的。如何解决这个问题？通过 `Embedding -> Pooling + Attention` 。

0x06 Embedding

深度学习在推荐、搜索领域的运用，是围绕着稀疏的ID类特征所展开的，其主要方法就是Embedding。变ID类特征的“精确匹配”为“模糊查找”，以增强扩展。即将高维、稀疏categorical/id类特征通过embedding映射成一个低维、稠密向量。

6.1 特点

Embedding层特点如下：

- 深度学习在推荐系统中的应用，比如各种NN，各种FM 都是以embedding为基础的；
- 高维、稀疏的categorical/id类特征是推荐系统中的一等公民；
- 在Embedding层中，每一个特征域都对应着一个Embedding矩阵；
- embedding的作用是将原来高维、稀疏的categorical/id类特征的“精确匹配”，变为向量之间的“模糊查找”，从而提高了可扩展性；
- 推荐系统中的Embedding与NLP中的Embedding也有不同。
- NLP中，一句话的一个位置上只有一个词，所以Embedding往往变成了：从Embedding矩阵抽取与词对应的行上的行向量；
- 推荐系统中，一个Field下往往有多个Feature，Embedding是将多个Feature Embedding合并成一个向量，即所谓的**Pooling**。比如某个App Field下的Feature有"微信:0.9，微博:0.5，淘宝:0.3"，所以得到 `Embedding = 0.9 * 微信向量 + 0.5 * 微博向量 + 0.3 * 淘宝向量` ；

6.2 变长特征

MLP只能接受固定长度的输入，但是每个用户在一段时间内的商品点击序列长度可能会不同，属于变长特征，那么该如何处理这样的变长特征？

一般来说是由 Pooling 层来处理，下面就让我们看看Pooling层。

0x07 Pooling 层

Pooling的作用是把embedding向量转化为一个固定长度的向量，解决维度不定的问题。

7.1 Pooling作用

用户有多个兴趣爱好，这导致两个问题：

- 表达用户兴趣时，用户的**历史行为往往涉及多个categorical/id特征**，比如点击过的多个商品、看过的多个视频、输入过的多个搜索词，这就涉及了多个good_id，shop_id。
- 不同的用户有不同数量的历史行为，即multi-hot行为特征的向量会导致所产生的embedding向量列表的长度不同，而全连接需要固定长度的输入。

为了降低纬度并使得商品店铺间的算术运算有意义，我们先对id特征进行Embedding嵌入。

那么如何对用户多种多样的兴趣建模？我们把这些id特征embedding之后的多个低维向量（embedding向量列表），“合并”成一个向量，作为用户兴趣的表示。

因为全连接需要固定长度的输入，所以我们需要“合并”成一个固定长度向量，这样才能喂入DNN。

这个“合并”就是所谓**Pooling**。

7.2 实现方式

围绕着这个Pooling过程，各家有各家的高招：

- Youtube DNN这篇论文中，Youtube的做法最简单、直观，就是将用户看过的视频embedding向量、搜索过的关键词embedding向量，做一个**简单的平均**。
- Neural Factorization Machine中，将n个(n=特征数)k维向量压缩成一个k维向量，取名为bi-interaction pooling。既完成pooling，也实现了特征间的二阶交叉。

- DIN用各embedding向量的加权平均实现了pooling，而”权重”由attention机制计算得到。

- **基于深度学习的文本分类，同样面临着如何将一段话中的多个词向量压缩成一个向量来表示这段话的问题。常用的方法，就是将多个词向量喂入RNN，最后一个时刻RNN的输出向量就代表了多个词向量的“合并”结果。**显然，DIEN则借鉴了这一思路，并且改造了GRU的构造，利用attention score来控制门。

7.3 DNN

DNN base模型采用pooling的方式，一般有两种方法，**求和池化**（sum pooling，各个对应元素进行累加）和**平均池化**（average pooling，各个对应元素求平均）。然后将所有向量连接在一起（concatenate），以获得实例的总体表示向量。

- 求和就是对多个商品的embedding，在每个对应的维度上做求和。例如，点击序列有10个商品，那么就有10个商品的embedding，假设商品的embedding维度是16，那么分别在第1到16维上，对10个值求和。
- 平均就是对多个embedding，在每个对应的维度上求平均。不管用户点击过多少个商品，经过pooling之后，得到的最终表示向量embedding和每个商品的embedding维度都是相同的。

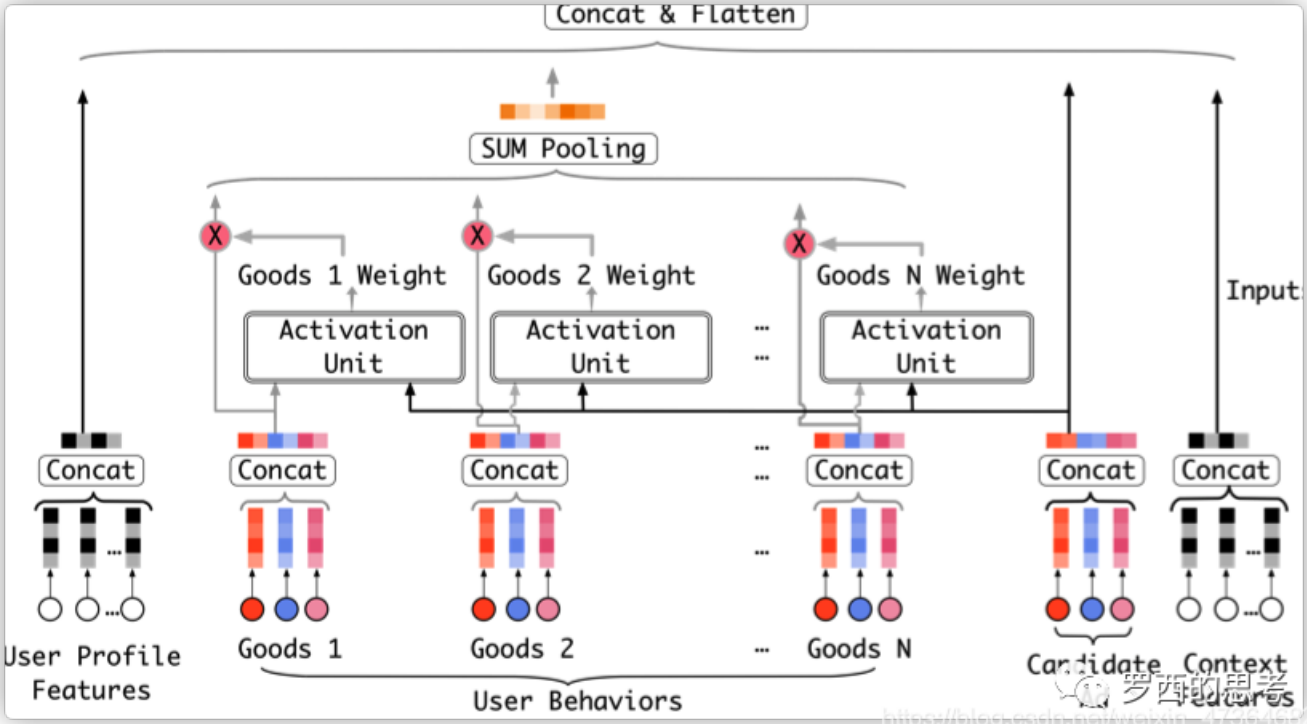
base模型对于任何要预测的candidate，不管这个candidate是衣服，电子产品等，用户的表示向量都是确定的、不变的，对于任何candidate都无差别对待。

7.4 DIN

回到阿里的展示广告系统，如架构图所示，每个商品有3个特征域，包括商品自身，商品类别，商品所属的商铺。对于每个商品来说，3个特征embedding拼接之后才是商品的表示向量。

对商品序列做pooling，架构图中采用的是求和的方式，pooling之后得到用户行为序列的表示向量。然后再和其他的特征embedding做拼接，作为MLP的输入。

MLP输入端的整个embedding向量，除了candidate的embedding部分，其余的embedding部分可以视为用户的表示向量。



在这里插入图片描述

仔细的研究下Base Model中Pooling Layer就会发现，Pooling操作损失了很多信息。

所以DIN 使用 Pooling（weighted sum）对Diversity建模，因为直接sum体现不出差异多样性，加权可以。

即DIN用各embedding向量的加权平均实现了pooling，而"权重"由attention机制计算得到。

0x08 Attention机制

Attention机制简单的理解就是，针对不同的广告，用户历史行为与该广告的权重是不同的。假设用户有ABC三个历史行为，对于广告D，那么ABC的权重可能是0.8、0.1、0.1；对于广告E，那么ABC的权重可能是0.3、0.6、0.1。这里的权重，就是Attention机制即架构图中的Activation Unit所需要学习的。

DIN模型其实就是在DNN基础上加了attention。通过Attention来实现Pooling，使用户兴趣的向量表示，根据候选物料的不同而不同，实现**用户兴趣的"千物千面"**。

模型的目标：基于用户历史行为，充分挖掘用户兴趣和候选广告之间的关系。用户是否点击某个广告往往是基于他之前的部分兴趣，这是应用Attention机制的基础。因为无论是用户兴趣行为，还是候选广告都会被映射到**Embedding空间**中。所以他们两者的关系，是在Embedding空间中学习的。

8.1 问题

DIN的attention机制部分是**为了**用一个 fix length 的 vector 刻画用户面对不同的商品展现出不同的兴趣，这个点看起来很简单，但是实际比较困难。

- 传统DNN模型为了得到一个固定长度的Embedding Vector表示，原来的做法是在 Embedding Layer 后面增加一个 Pooling Layer 。Pooling可以用sum或average。最终得到一个固定长度的 Embedding Vector ，是用户兴趣的一个抽象表示，常被称作 User Representation 。缺点是会损失一些信息。
- 用户Embedding Vector的维度为k，它最多表示k个相互独立的兴趣爱好。但是用户的兴趣爱好远远不止k个，怎么办?
- 传统DNN模型在 Embedding Layer -> Pooling Layer 得到用户兴趣表示的时候，也没有考虑用户与广告之间的关系，即不同广告之间的权重是一致的。这样传统的预估方法在一个user面对不同商品（广告）时用一个同样的vector来表达这个user。如果在这种情况下要想表达多样的兴趣，最简单的方案是增加user vector的维度，然而这会带来overfitting和计算压力。

所以DIN用类似attention的机制试图解决这个问题。

8.2 注意力机制

注意力机制顾名思义，就是模型在预测的时候，对用户不同行为的注意力是不一样的，“相关”的行为历史看重一些，“不相关”的历史甚至可以忽略。即对于不同的特征有不同的权重，这样某些特征就会主导这一次的预测，就好像模型对某些特征pay attention。

这样的思想反应到模型中也是直观的。比如在视频推荐模型中，DIN可以通过增加用户的历史行为feature：用户观看的近20个show_id，近20个video_id，然后使用attention网络，最后与其它非历史行为feature在MLP中汇合。

DIN利用attention机制去更好的建模局部激活。在得到用户兴趣表示时赋予不同的历史行为不同的权重，即通过 Embedding Layer -> Pooling Layer+attention 实现局部激活。从最终反向训练的角度来看，就是根据当前的候选广告，来反向的激活用户历史的兴趣爱好，赋予不同历史行为不同的权重。

DIN给出的方案是：不再用一个点来表示用户兴趣，而是通过用一个在不同时刻不一样的分布表示：分布可以是多峰的，可以表达每个人有多个兴趣。一个峰就表示一个兴趣，峰值的大小表示兴趣强度。那么针对不同的候选广告，用户的兴趣强度是不同的，也就是说随着候选广告的变

化，用户的兴趣强度不断在变化。因为用户兴趣是一个多峰的函数，这样即使在低维空间，也可以获得几乎无限强的表达能力。

换句话说：**假定用户兴趣表示的Embedding Vector是Vu，候选广告的是Va，那么Vu是Va的函数。** 也就是说，同意用户针对不同的广告有不同的用户兴趣表示（嵌入向量不同）。

其中：

- Vi表示behavior id i的嵌入向量，比如good_id,shop_id等。
- Vu是所有behavior ids的加权和，表示的是用户兴趣。
- 候选广告影响着每个behavior id的权重，也就是Local Activation。
- 权重表示的是：每一个behavior id针对当前的候选广告Va，对总的用户兴趣表示的Embedding Vector的贡献大小。在实际实现中，权重用激活函数Dice的输出来表示，输入是Vi和Va。

8.3 实现

DIN中并不能直接用attention机制。因为对于不同的候选广告，用户兴趣表示（embedding vector）应该是不同的。

Local Activation Unit 借鉴了NMT（Neural Machine Translation）中的attention机制，实现了自己的Attention机制。Local Activation学习候选广告和用户历史行为的关系，并给出候选广告和各个历史行为的相关性程度 （即权重参数），再对历史行为序列进行加权求和，最终得到用户兴趣的特征表达。也就是说用户针对不同的广告表现出不同的兴趣表示，即使历史兴趣行为相同，但是各个行为的权重不同。

DIN 在pooling的时候，与candidate相关的商品权重大一些，与candidate不相关的商品权重小一些，**这就是一种Attention的思想**。将candidate与点击序列中的每个商品发生交互来计算attention分数。具体计算输入包括商品和candidate的embedding向量，以及两者的外积。对于不同的candidate，得到的用户表示向量也不同，具有更大的灵活性。

DIN中，对于候选广告， 根据local activation unit计算出的用户兴趣向量为：

$$\boldsymbol{v}_U(A) = f(\boldsymbol{v}_A, \boldsymbol{e}_1, \boldsymbol{e}_2, .., \boldsymbol{e}_H) = \sum_{j=1}^{H} a(\boldsymbol{e}_j, \boldsymbol{v}_A)\boldsymbol{e}_j = \sum_{j=1}^{H} w_j \boldsymbol{e}_j, \quad (3)$$

在这里插入图片描述

其中，

- ei 表示用户U历史行为序列embedding向量，比如good_id,shop_id等，长度为H；
- Vu 表示用户所有行为embedding向量的加权和，表示用户的兴趣；
- Va 表示广告 A 的mbedding 向量;
- wj 表示ej的权重；
- 权重表示的是：每一个behavior id针对当前的候选广告Va，对总的用户兴趣表示的Embedding Vector的贡献大小。
- 在实现中，权重wj 用函数去拟合，通过Activation Unit计算得出，用激活函数Dice的输出来表示，表示为g(Vi,Va)，输入是Vi和Va；
- 候选广告影响着每个behavior id的权重，也就是Local Activation；
- a （ . ） 表示一个feed-forward network，其输出作为local activation的权值，与用户向量相乘；

在这种计算方式下，最终的用户 U 的兴趣向量会根据不同的广告 A 而变化。这就是 “**用户兴趣的千物千面**”。比如，一个用户之前买过奶粉与泳衣，当展示给她泳镜时，显然更会唤起她买过的泳衣的记忆；而当展示给她尿不湿时，显然更唤起她买过的奶粉的记忆。

DIN attention机制中，用户兴趣向量 Vu 是历史上接触过的item embedding向量的加权平均，而第 i 个历史 item 的权重 Wi 由该历史 item 的 embedding 向量 Vi 与候选物料的 embedding 向量 Va 共同决定（函数g）。可见同一个用户当面对不同候选物料时，其兴趣向量也不相同，从而实现了“千物千面”。

DIN与base model的主要区别就在于激活单元上，这个结构通过计算广告的embedding与用户表现的embedding之间的相似度得到对应的权重，后对表现序列进行权重求和，取得了不俗的表现。

8.4 归一化

一般来说，做attention的时候，需要对所有的分数通过softmax做归一化，这样做有两个好处，一是保证权重非负，二是保证权重之和为1。

但是在DIN的论文中强调，不对点击序列的attention分数做归一化，直接将分数与对应商品的embedding向量做加权和，目的在于保留用户的兴趣强度。例如，用户的点击序列中90%是衣服，10%是电子产品，有一件T恤和一部手机需要预测CTR，那么T恤会激活大部分的用户行为，使得根据T恤计算出来的用户行为向量在数值上更大。

0x09 评价指标

评价标准是阿里自己提出的GAUC。并且实践证明了GAUC相比于AUC更加稳定、可靠。

AUC表示正样本得分比负样本得分高的概率。在CTR实际应用场景中，CTR预测常被用于对每个用户候选广告的排序。但是不同用户之间存在差异：有些用户天生就是点击率高。以往的评价指标对样本不区分用户地进行AUC的计算。论文采用的GAUC实现了用户级别的AUC计算，**在单个用户AUC的基础上，按照点击次数或展示次数进行加权平均，消除了用户偏差对模型的影响**，更准确的描述了模型的表现效果。

0x10 Adaptive Regularization

由于深度模型比较复杂，输入又非常稀疏，导致参数非常多，非常容易过拟合。

CTR中输入稀疏而且维度高，已有的L1 L2 Dropout防止过拟合的办法，论文中尝试后效果都不是很好。用户数据符合 长尾定律 long-tail law ，也就是说很多的feature id只出现了几次，而一小部分feature id出现很多次。这在训练过程中增加了很多噪声，并且加重了过拟合。

对于这个问题一个简单的处理办法就是：人工的去掉出现次数比较少的feature id。缺点是：损失的信息不好评估；阈值的设定非常的粗糙。

DIN给出的解决方案是：

1. 针对feature id出现的频率，来自适应的调整他们正则化的强度；
2. 对于出现频率高的，给与较小的正则化强度；
3. 对于出现频率低的，给予较大的正则化强度。

对L2正则化的改进，在进行SGD优化的时候，每个mini-batch都只会输入部分训练数据，反向传播只针对部分非零特征参数进行训练，添加上L2之后，需要对整个网络的参数包括所有特征的embedding向量进行训练，这个计算量非常大且不可接受。论文中提出，在每个mini-batch中只对该batch的特征embedding参数进行L2正则化。

0x11 总结

对论文总结如下：

1. 用户有多个兴趣爱好，访问了多个good_id，shop_id。为了降低纬度并使得商品店铺间的算术运算有意义，我们先对其进行Embedding嵌入。那么我们如何对用户多种多样的兴趣建模那？使用**Pooling对Embedding Vector求和或者求平均**。同时这也解决了不同用户输入长度不同的问题，得到了一个固定长度的向量。这个向量就是用户表示，是用户兴趣的代表。

2. 但是，直接求sum或average损失了很多信息。所以稍加改进，针对不同的behavior id赋予不同的权重，这个权重是由当前behavior id和候选广告共同决定的。这就是Attention机制，实现了Local Activation。
3. DIN使用*activation unit*来捕获local activation的特征，使用*weighted sum pooling*来捕获diversity结构。
4. 在模型学习优化上，DIN提出了*Dice*激活函数、自适应正则，显著的提升了模型性能与收敛速度。

0xFF 参考

用NumPy手工打造 Wide & Deep

看Google如何实现Wide & Deep模型(1)

看Youtube怎么利用深度学习做推荐

也评Deep Interest Evolution Network

从DIN到DIEN看阿里CTR算法的进化脉络

第七章 人工智能，7.6 DNN在搜索场景中的应用(作者：仁重)

#Paper Reading# Deep Interest Network for Click-Through Rate Prediction

【paper reading】Deep Interest Evolution Network for Click-Through Rate Prediction

也评Deep Interest Evolution Network

论文阅读：《Deep Interest Evolution Network for Click-Through Rate Prediction》

【论文笔记】Deep Interest Evolution Network(AAAI 2019)

【读书笔记】Deep Interest Evolution Network for Click-Through Rate Prediction

DIN(Deep Interest Network):核心思想+源码阅读注释

计算广告CTR预估系列(五)--阿里Deep Interest Network理论

CTR预估之Deep Interest NetWork模型原理详解