

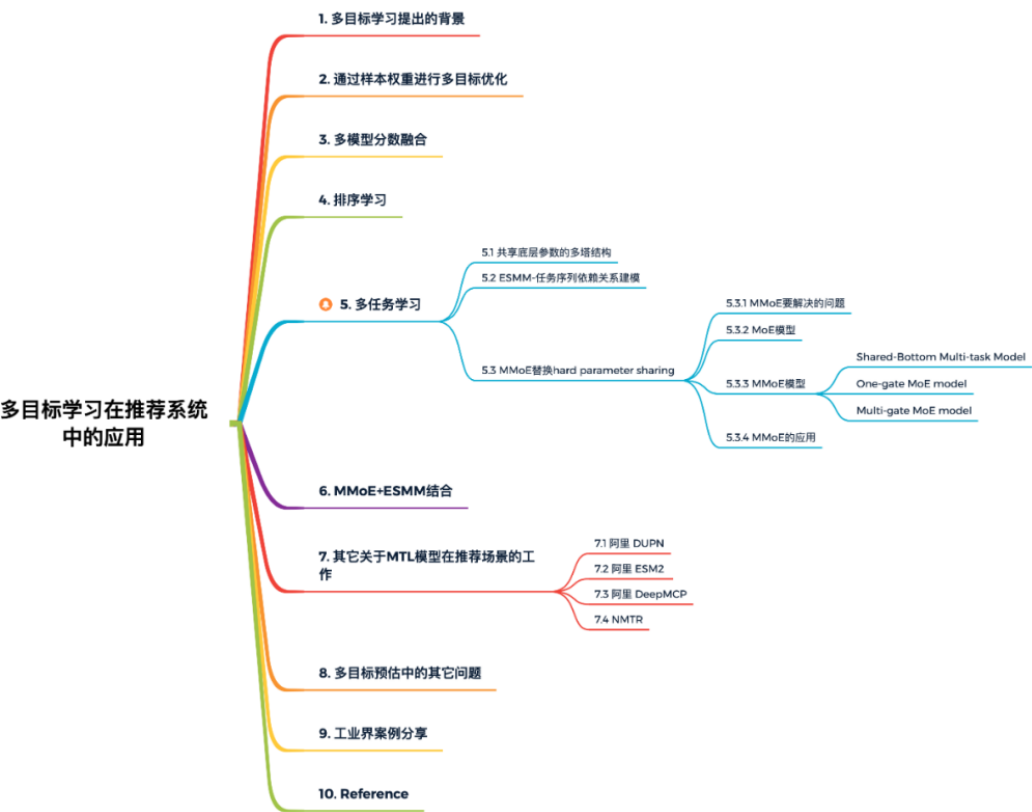
多目标学习在推荐系统中的应用

原创 Microstrong Microstrong 11月4日

收录于话题
#推荐系统原理解析与实践部署

8个

本文概览：



1. 多目标学习提出的背景

一般来说在搜索和推荐等信息检索场景下，最基础的一个目标就是用户的 CTR，即用户看见了一篇内容之后会不会去点击阅读。但其实用户在产品上的行为是多种多样的。比如在微信的订阅号中，用户可以对某个内容进行点赞，可以收藏这个内容，可以把它分享出去，甚至某篇文章如果他觉得比较符合他的兴趣，也可以进行留言。

虽然可以对用户的 CTR 进行单个目标的优化，但是这样的做法也会带来负面影响：靠用户点击这个行为推荐出来的内容并不一定是用户非常满意的内容，比如有人可能看到一些热门的内容就会去点击，或者看到一些阅读门槛低的内容，像一些引发讨论的热点事件、社会新闻，或者是一些轻松娱乐的内容，用户也会点击。这样造成的后果就是：CTR 的指标非常高，但是用户接收到的推荐结果并不是他们最满意的。

如果我们深入思考会发现，用户的每种行为一定程度上都代表了某个内容是否能满足他不同层面的需求。比如说点击，代表着用户在这个场景下，想要看这个内容；赞同，代表用户认为这个内容其实写的很不错；收藏，代表这个内容对用户特别有用，要把它收藏起来，要仔细的去看一看；分享，代表用户希望其他的人也能看到这个内容。

而单目标 CTR 优化到了一个比较高的点之后，用户的阅读量虽然上去了，但是其他的各种行为（收藏、点赞、分享等等）是下降的。这个下降代表着：用户接收到太多的东西是他认为不实用的。

于是，推荐系统研究者开始思考：能不能预估用户在其他行为上的概率？这些概率实际上就是模型要学习的目标，多种目标综合起来，包括阅读、点赞、收藏、分享等等一系列的行为，归纳到一个模型里面进行学习，这就是推荐系统的多目标学习。

做推荐算法肯定绕不开多目标。点击率模型、时长模型和完播率模型是大部分信息流产品推荐算法团队都会尝试去做的模型。单独优化点击率模型容易推出来标题党，单独优化时长模型可能推出来的都是长视频或长文章，单独优化完播率模型可能短视频短图文就容易被推出来，所以多目标就应运而生。

多目标排序问题的解决方案，大概有以下四种：通过改变样本权重、多模型分数融合、排序学习（Learning To Rank, LTR）、多任务学习（Multi-Task Learning, MTL）。

2. 通过样本权重进行多目标优化

如果主目标是点击率，分享功能是我们希望提高的功能。那么点击和分享都是正样本（分享是点击行为的延续），分享的样本可以设置更高的样本权重。在模型训练计算梯度更新参数时，梯度要乘以权重，对样本权重大的样本给予更大的权重。

因此，样本权重大的样本，如果预测错误就会带来更大的损失。通过这种方法能够在优化某个目标（点击率）的基础上，优化其他目标（分享率）。实际AB测试会发现，这样的方法，目标A会受到一定的损失换取目标B的增长。通过线上AB测试和样本权重调整的联动，可以保证在可接受的A目标损失下，优化目标B，实现初级的多目标优化。

优点：

- 模型简单，仅在训练时通过梯度上乘样本权重实现对某些目标的放大或者减弱。
- 带有权重样本的模型和线上的base模型完全相同，不需要架构的额外支持，可以作为多目标的第一个模型尝试。

缺点：

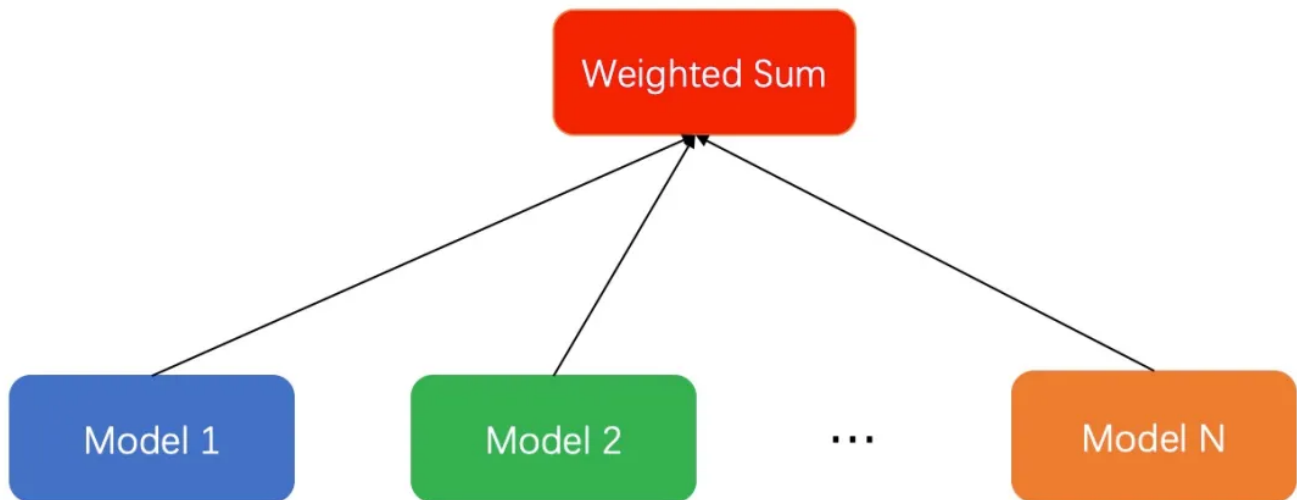
- 本质上并不是对多目标的建模，而是将不同的目标折算成同一个目标。样本的折算权重需要根据AB测试才能确定。比如认为一次分享算两次点击，在视频中停留了2分钟等价于3次对视频的点击行为等，这里面的数字需要根据线上评估指标测试出来。
- 从原理上讲无法达到最优，多目标问题本质上是一个帕累托寻找有效解的过程。

3. 多模型分数融合

刚开始做多目标的时候，一般针对每一个目标都单独训练一个模型，单独部署一套预估服务，然后将多个目标的预估分融合后排序。这样能够比较好的解决推荐过程当中的一些负面case，在各个指标之间达到一个平衡，提升用户留存。但是同时维护多个模型成本比较高，首先是硬件上，需要多倍的训练、PS和预估机器，这是一笔不小的开销，然后是各个目标的迭代不好协同，比如新上了一批好用的特征，多个目标都需要重新训练实验和上线，然后就是同时维护多个目标对相关人员的精力也是一个比较大的消耗。

具体的做法为：假如我们有多个优化的目标，每个优化的目标都有一个独立的模型来优化。可以根据优化目标的不同，采用更匹配的模型。如视频信息流场景中，我们用分类模型优化点击率，用回归模型优化停留时长。不同的模型得到预测的score之后，通过一个函数将多个目标融合在一起。最常见的是weighted sum融合多个目标，给不同的目标分配不同的权重。当然，融合的函数可以有很多，比如连乘或者指数相关的函数，这里和业务场景和目标的含义强相关，可以根据自己的实际场景探索。这种做法优点：模型简单；缺点是：（1）线上serving部分需要额外的时间开销，通常我们采用并行的方式请求多个模型进行融合。（2）多个模型之间相互独立，不能互相利用各自训练的部分作为先验，容易过

拟合。



4. 排序学习

多模型分数融合通过计算推荐物的综合得分来排序（point-wise），其目的是为了推荐物品。因此，也可以直接预测物品两两之间相对顺序的问题（pair-wise）来解决多目标学习的问题，常用的算法比如 BPR，还可以预测物品序列之间的得分情况（list-wise）来解决多目标学习的问题。具体可以看看相关算法的介绍。

举一个用pair-wise做视频推荐的例子，用户观看很长时间的 i 视频，点击了 j 视频，那么我们觉得观看比点击重要。用户 u 在视频 i 和视频 j 之间的偏好和其他的视频无关。为了便于描述，用 $>_u$ 符号表示用户 u 的偏好，上面的 $\langle u, i, j \rangle$ 可以表示为： $i >_u j$ 。有了相对顺序后可以使用排序学习的方式，融合多个目标，不需要训练多个模型。

优点：

- 优化了目标排序，不需要设计复杂的超参数，能取得比排序好的效果。
- 本身就是单个模型有多个目标，线下好训练，线上服务压力小。

缺点：

- 有些相对顺序不好构造，训练样本中没有的关系，在预测时可能存在。
- 样本数量增大，训练速度变慢，需要构造的情况多。
- 样本的不平衡性会被放大。举例：有的用户有十次点击，有的只有一次，在构造的时候十次的会构造更多的样本，一次的就吃亏。

当然，通过LTR（Learning To Rank）方法优化Item的重要性可以解决多目标要解决的问题，但是由于工程实现、推荐框架调整等方面的困难，相关方法在实际应用中比较少。下面我们重点梳理一下一个模型预估多个目标的多任务学习模型的发展进程。

5. 多任务学习

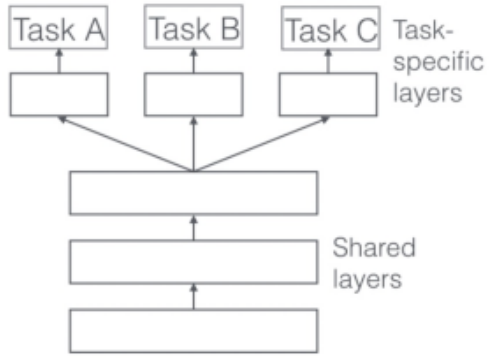
5.1 共享底层参数的多塔结构

2017年的这篇综述文章《An Overview of Multi-Task Learning in Deep Neural Networks》将多目标的方法分成了两类。第一类是Hard parameter sharing的方法，将排序模型底层的全连接层进行共享，学习共同的模式，上层用一些特定的全连接层学习任务特定的模式。这种方法非常经典，效果还是不错的，美团、知乎在18年分享的文章《美团“猜你喜欢”深度学习排序模型实践》、《知乎推荐页Ranking经验分享》中使用的都是这种方法。

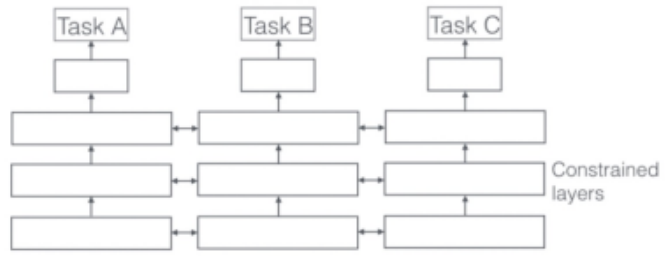
【相关链接】

- Ruder S. An overview of multi-task learning in deep neural networks[J]. arXiv preprint arXiv:1706.05098, 2017.
- 美团“猜你喜欢”深度学习排序模型实践，地址：
<https://tech.meituan.com/2018/03/29/recommend-dnn.html>
- 「回顾」知乎推荐页Ranking经验分享，地址：<https://mp.weixin.qq.com/s/GUMz-HfbjQvzdGVQkKz4zA>

Two MTL methods for Deep Learning



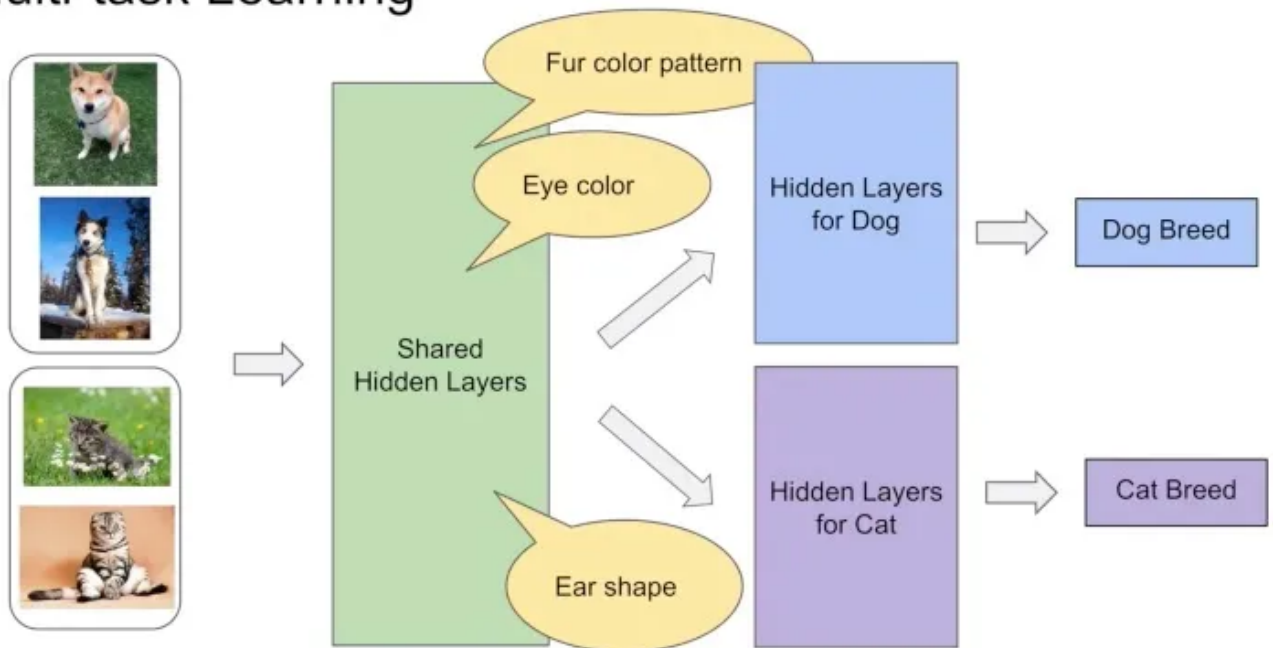
Hard parameter sharing



Soft parameter sharing

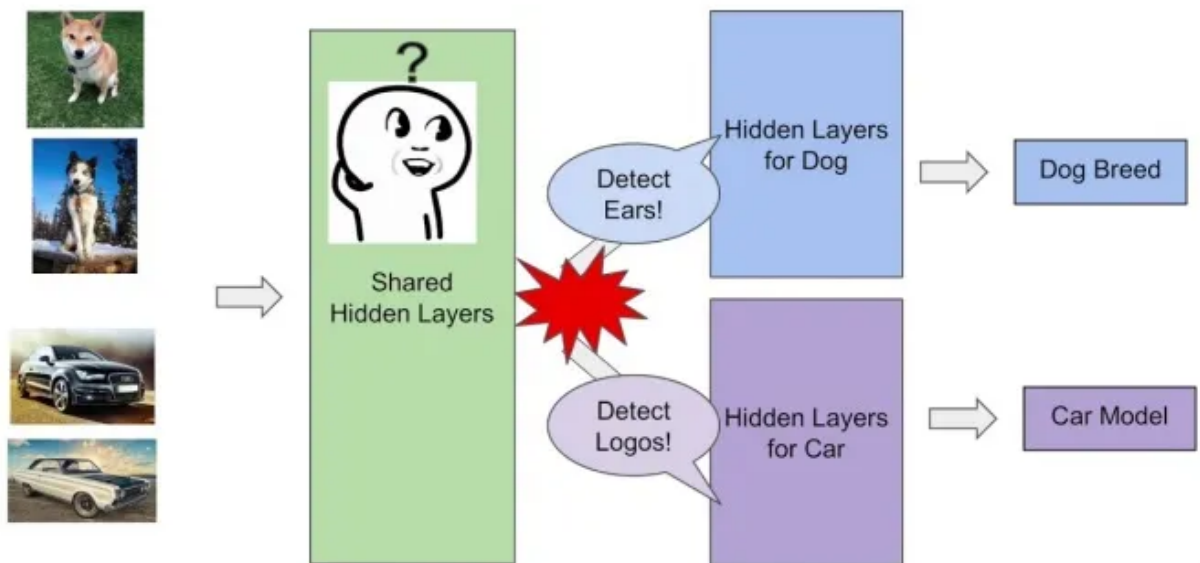
这种方法最大的优势是Task越多，单任务更加不可能过拟合，劣势是底层强制shared layers难以学得适用于所有任务的表达。尤其是两个任务很不相关的时候。如下图中的例子，假设你的多目标预估是同时做猫狗的分类，那么底层的shared layers学到的可能是关于眼睛、耳朵、颜色的一些共同模式。

Multi-task Learning



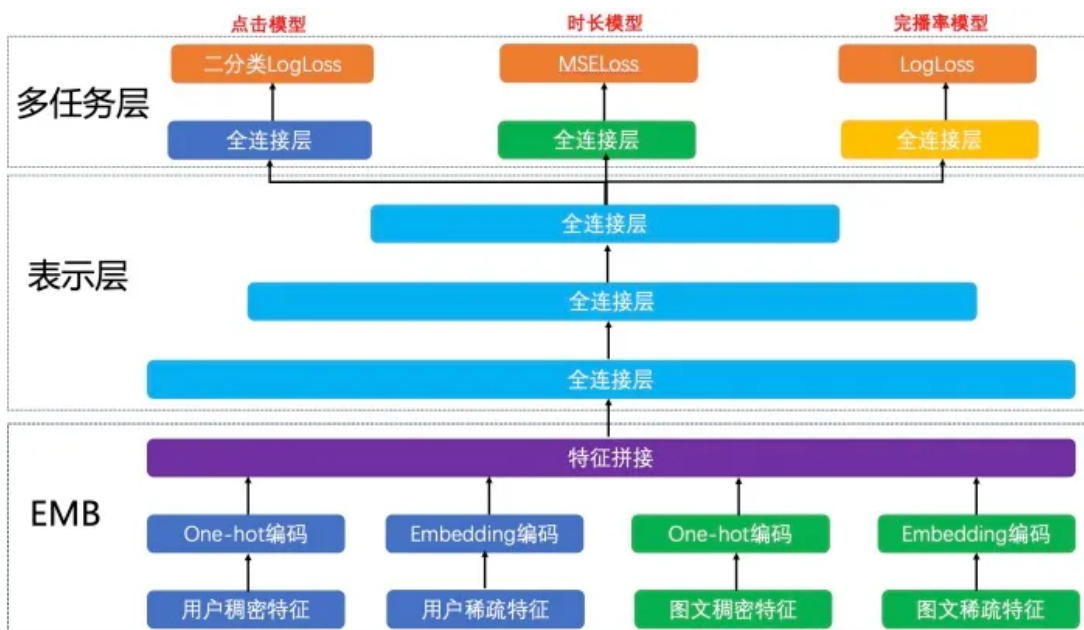
而如果你的任务是做一个狗的分类，同时做汽车的分类，采用Hard parameter sharing的方法，底层的shared layers很难学到两者共同的模式，多目标任务也很难学好。

Task conflicts in multi-task learning



谷歌在MMoE论文中的实验显示，任务相关性越高，模型的loss可以降到更低。当两个任务相关性没有那么好时（例如，推荐系统排序中的点击率和互动率、点击率和停留时长等），Hard parameter sharing 的模式就不是那么适用了，会损害到一些效果。对应的，Soft parameter sharing的方式可能更加适合。

下图展示的是在推荐中共享Embedding多塔结构的例子。根据业务目标，我们把点击率、时长和完播率拆分出来，形成三个独立的训练目标，分别建立各自的Loss Function，作为对模型训练的监督和指导。此模型的EMB部分和表示层作为共享层，点击任务、时长任务和完播率任务共享其表达，并在BP阶段根据三个任务算出的梯度共同进行参数更新。在表示层的最后一个全连接层进行拆分，单独学习对应Loss的参数，从而更好地专注于拟合各自Label的分布。线上预测时，我们将点击模型的output、时长模型的output和完播率模型的output做一个线性融合。



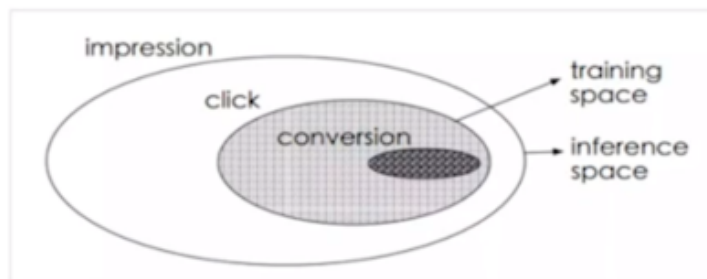
虽然解决了

一部分问题，但是把多个模型融合在一起，通过一个模型去学习一个目标的方式仍然存在。首先，目标之间的相关性决定了这个模型学习的上限能有多少。比如：如果一个模型中点赞和点击是完全耦合的，那么这个模型在学习点赞的过程中，也就学习了点击。但是对用户来讲，它的意义是不一样的，这并不是一个完全耦合的系统。在这个学习任务下，如果去共享底层网络参数的话，可能会造成底层的每个目标都能学习一点，但是每个目标学习的都不够充分，这是多目标学习系统实现的一个难点。

后来，阿里妈妈的Xiao Ma等人发现，在推荐系统中不同任务之间通常存在一种序列依赖关系。例如，电商推荐中的多目标预估经常是CTR和CVR，其中转化这个行为只有在点击发生后才会发生。这种序列依赖关系其实可以被利用，来解决一些任务预估中存在的样本选择偏差(Sample Selection Bias,SSB)和数据稀疏性(Data Sparsity,DS)问题。因此提出Entire Space Multi-Task Model (ESMM)。

简单介绍一下什么是SSB问题和DS问题。SSB问题：后一阶段的模型基于上一阶段采样后的样本子集进行训练，但是最终在全样本空间进行推理，带来严重的泛化性问题。DS问题：后一阶段的模型训练样本通常远小于前一阶段任务。

5.2 ESMM-任务序列依赖关系建模



本模型首次将样本分为：

曝光 -> 点击 -> 转化

三步的行为链，并引入了浏览转化率 (pCTCVR) 的概念，即为在浏览的条件下既点击又转化的概率，即：

$$p(\text{conversion}=1, \text{click}=1 | \text{impression}) = p(\text{click}=1 | \text{impression}) * p(\text{conversion}=1 | \text{click}=1, \text{impression})$$

pCTCVR

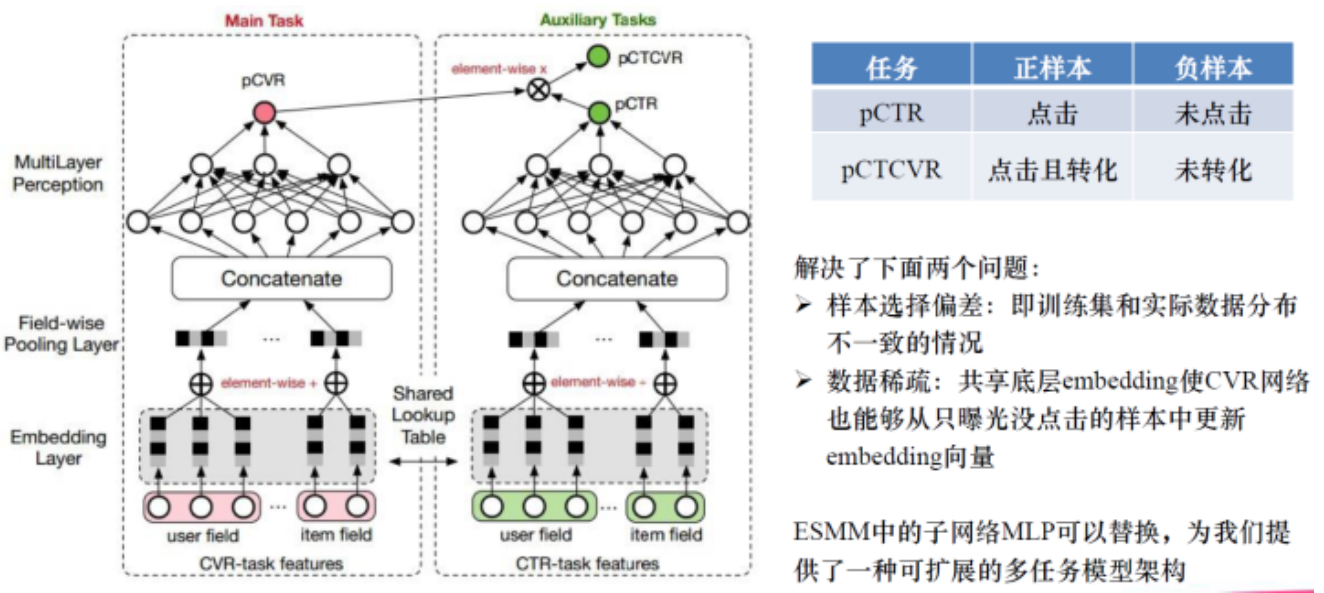
pCTR

pCVR

ESMM模型首次把样本空间分成了三大部分，从曝光到点击再到转化的三行为链，并引入了浏览转化率 (pCTCVR) 的概念。我们正常做CVR任务的时候，默认只在点击的空间上来做，认为曝光、点击并转化了就是正样本，曝光、点击并未转化为负样本。如果这样想的话，样本全空间只有点击的样本，而没有考虑到未点击的样本。ESMM论文就提出曝光点击、曝光不点击以及点击之后是否转化所有的这些样本都考虑进来，提出三步的行为链

如下公式：

$$p(\text{conversion} = 1, \text{click} = 1 | \text{impression}) = p(\text{click} = 1 | \text{impression}) * p(\text{conversion} = 1 | \text{click} = 1, \text{impression})$$



这就是浏览转化率公式，并提出了上图所示的网络模型，一个双塔模型，它们是共享底层Embedding的，只是上面的不一样，一个用来预测CVR，这个可以在全样本空间上进行训练，另一个是用来预测CTR，CTR是一个辅助任务。最后的pCTCVR可以在全样本空间中训练。ESMM模型是一个双塔、双任务的模式在全样本空间上进行训练。这样训练的好处可以解决两个问题：一是，样本选择的问题，CVR是在点击的基础上进行训练，训练集只有点击的，实际数据可能有曝光点击和曝光未点击的数据，我们往往把曝光未点击的数据给忽略了，这样就造成了样本选择偏差，训练集和实际数据分布不一致的情况。二是，解决数据稀疏的问题。因为我们现在在全样本空间上进行训练，不是只在点击的样本上进行训练，所以样本就多了很多，所有样本可以进行辅助更新CVR网络中的Embedding，这样Embedding向量就会训练的更加充分。ESMM还提出子网络MLP可以替换，为我们提供了一种可扩展的多任务模型架构。

ESMM是一种较为通用的任务序列依赖关系建模的方法，除此之外，阿里的论文《Deep Bayesian Multi-Target Learning for Recommender Systems》中提出DBMTL模型、《Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction》中提出 ESM^2 模型，这些工作都属于任务序列依赖这一模式。

5.3 MMoE替换hard parameter sharing

阿里团队提出 ESMM 模型利用多任务学习的方法极大地提升了 CVR 预估的性能，同时解决了传统 CVR 模型预估的一些弊病。我们从模型的网络结构可以了解到，ESMM 是典型的 share-bottom 结构，即底层特征共享方式。这种多任务学习共享结构的一大特点是在任务之间都比较相似或者相关性比较大的场景下能带来很好的效果，而对于任务间差异比较大的场景，这种多任务学习共享结构就有点捉襟见肘了。

5.3.1 MMoE要解决的问题

说到底，多任务学习的本质就是共享表示以及相关任务的相互影响。通常，相似的子任务也拥有比较接近的底层特征，那么多任务学习中，他们就可以很好地进行底层特征共享；而对于不相似的子任务，他们的底层表示差异很大，在进行参数共享时很有可能会互相冲突或噪声太多，导致多任务学习的模型效果不佳。

实际的应用场景中，我们可能不止有像 CTR、CVR 这样的非常相关的子任务，还会遇到子任务间关系没那么紧密的多任务学习场景，而且很多情况下，你很难判断任务在数据层面是否是相似的。所以多任务学习如何在相关性不高的任务上获得好效果是一件很有挑战性也很有实际意义的事，这也是本小节所提到的模型“MMoE”主要解决的问题。

其实在 MMoE 之前，已经有一些其他结构来解决问题了，比如两个任务的参数不共用，而是对不同任务的参数增加 L2 范数的限制；或者对每个任务分别学习一套隐层然后学习所有隐层的组合。这些结构和 Shared-Bottom 结构相比，其构成的模型会针对每个任务添加更多参数以适应任务间差异，虽然能够带来一定的效果提升，但是增加了更多的参数也就意味着需要更大的数据样本来训练模型，而且这些方法会使模型变得更复杂，也不利于在真实生产环境中部署使用。相关内容在 MMoE 论文中的“2.1 Multi-task Learning in DNNs”小节中有提到，想详细了解可以看论文中给出的引用文章。

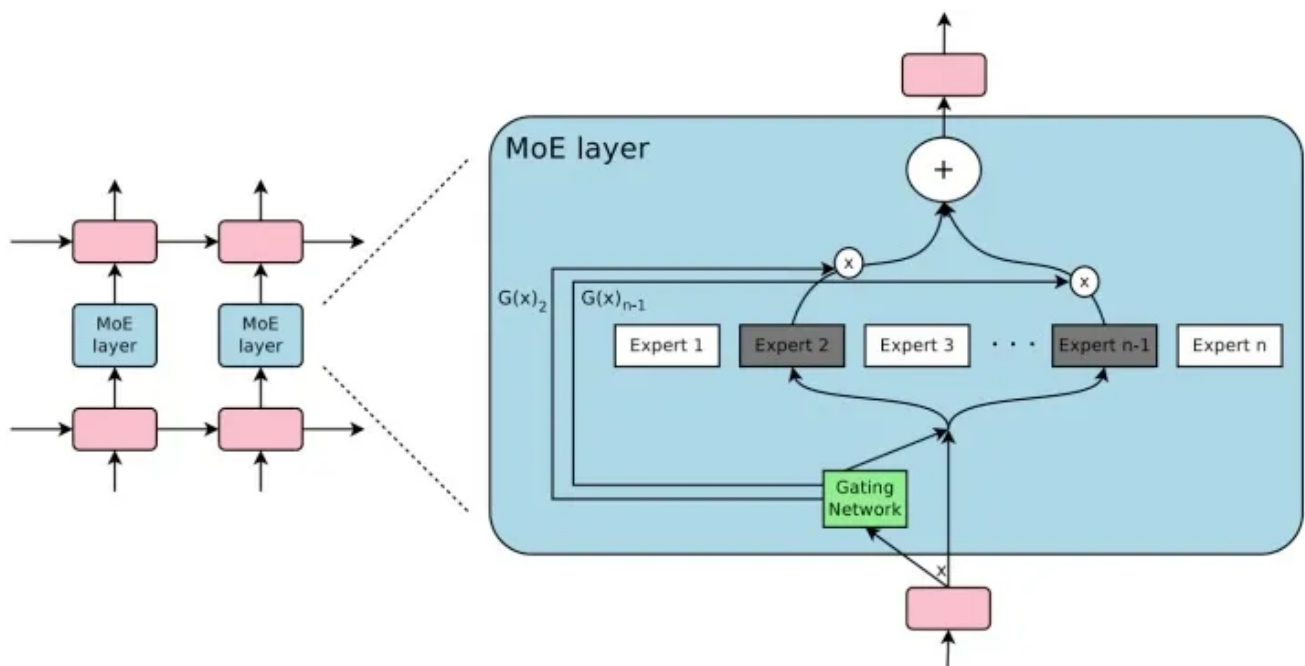
MMoE (Multi-gate Mixture-of-Experts) 是 Google 在 2018 年 KDD 上发表的论文《Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts》里提出的，它是一种新颖的多任务学习结构。MMoE 模型刻画了任务相关性，基于共享表示来学习特定任务的函数，避免了明显增加参数的缺点。广义上 MMoE 方法是 Soft parameter sharing 的一种。在这篇论文发表前，还有个 MoE 模型 (Mixture-of-Experts)，下面先介绍 MoE 模型。

5.3.2 MoE模型

早在 2017 年，谷歌大脑团队的两位科学家：大名鼎鼎的深度学习之父 *Geoffrey Hinton* 和谷歌首席架构师 *Jeff Dean* 在发表论文《Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer》中提出了“稀疏门控制的混合专家层”

（Sparsely-Gated Mixture-of-Experts layer, MoE），这里的 MoE 是一种特殊的神经网络结构层，结合了专家系统和集成思想在里面。

MoE 由许多“专家”组成，每个“专家”都有一个简单的前馈神经网络和一个可训练的门控网络（gating network），该门控网络选择“专家”的一个稀疏组合来处理每个输入，它可以实现自动分配参数以捕获多个任务可共享的信息或是特定于某个任务的信息，而无需为每个任务添加很多新参数，而且网络的所有部分都可以通过反向传播一起训练。MoE 结构图如下所示：



MoE 可以作为一个基本的组成单元，也可以是多个 MoE 结构堆叠在一个大网络中。比如一个 MoE 层可以接受上一层 MoE 层的输出作为输入，其输出作为下一层的输入使用。在谷歌大脑的论文中，MoE 就是作为循环神经网络中的一个循环单元。

MoE 神经网络结构有两个非常明显的好处：

（1）实现一种多专家集成的效果。

MoE 的思想是训练多个神经网络（也就是多个专家），每个神经网络（专家）通过门控网络（Gating NetWork）被指定应用于数据集的不同部分，最后再通过门控网络将多个专家的结果进行组合。单个模型往往善于处理一部分数据，不擅长处理另外一部分数据（在这部分数据上犯错多），而多专家系统则很好的解决了这个问题：系统中的每一个神经网络，也就是每一个专家都会有一个擅长的数据区域，在这组区域上该专家就是“权威”，要比其他专家表现得好。因此多专家系统是单一全局模型或者多个局部模型的一个很好的折

中，这样的网络结构能够处理更加复杂的数据分布，在相应的任务中，性能也会有很大的提升。

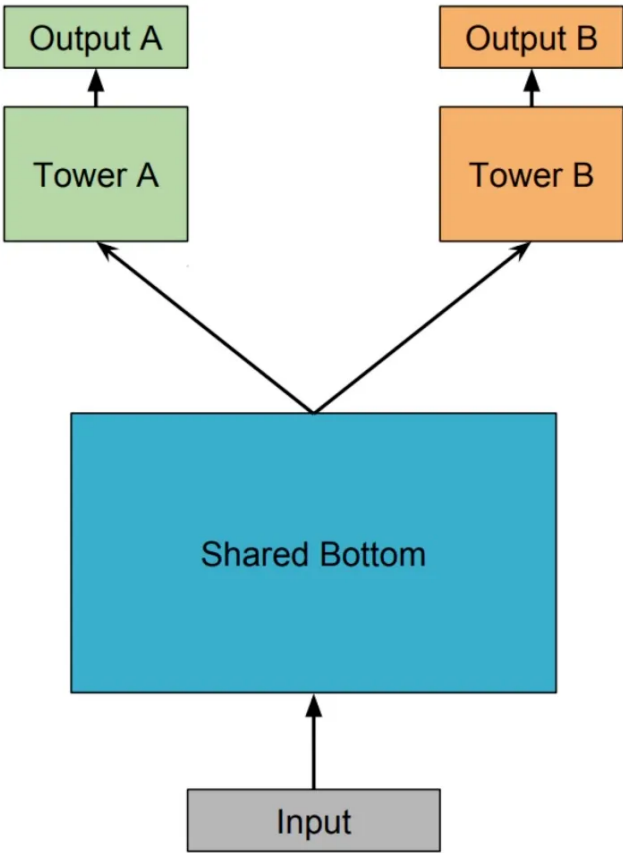
（2）只需增加很小的计算力，便能高效地提升模型的性能。

神经网络吸收信息的能力受其参数数量的限制。有人在理论上提出了条件计算（conditional computation）的概念，作为大幅提升模型容量而不会大幅增加计算力需求的一种方法。MoE 就是条件计算的一种实现，并在论文中证实，这种网络结构可实现在计算效率方面只有微小损失情况下，可以显着提高性能。

5.3.3 MMoE模型

有了以上信息，我们就很容易理解为什么要将 MoE 引入多任务学习中了，因为多专家集成的机制刚好可以用来解决多任务间的差异问题，并且还不会带来很大计算损耗。

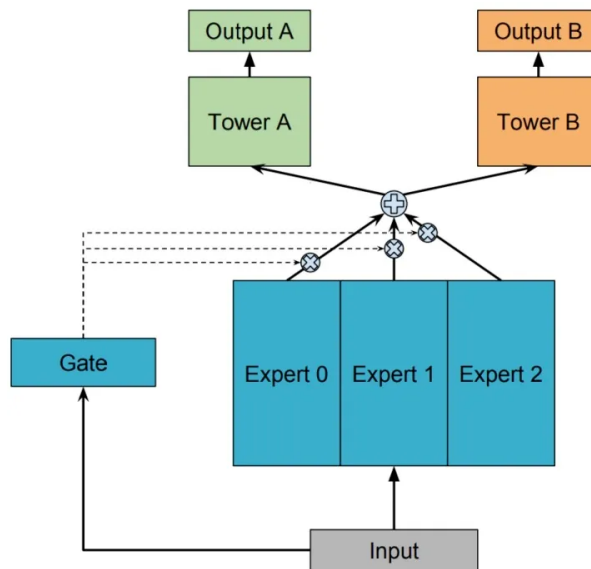
- Shared-Bottom Multi-task Model 多任务学习（Multi-Task Learning，MTL）中最经典的 Shared-Bottom DNN 网络结构，如下图所示：



Shared-Bottom 网络通常位于底部，表示为函数 f ，多个任务共用这一层。往上， K 个子任务分别对应一个 tower network，表示为 h^k ，每个子任务的输出 $y_k = h^k(f(x))$ 。

- One-gate MoE model

用一组由专家网络（expert network）组成的神经网络结构来替换掉 Shared-Bottom 部分（函数 f ），这里的每个“专家”都是一个前馈神经网络，再加上一个门控网络，就构成了 MoE 结构的 MTL 模型。因为只有一个门网络，所以在论文中，为了与 MMoE 对应，也称这种结构为 OMoE（One-gate Mixture-of-Experts），其结构如下图所示：



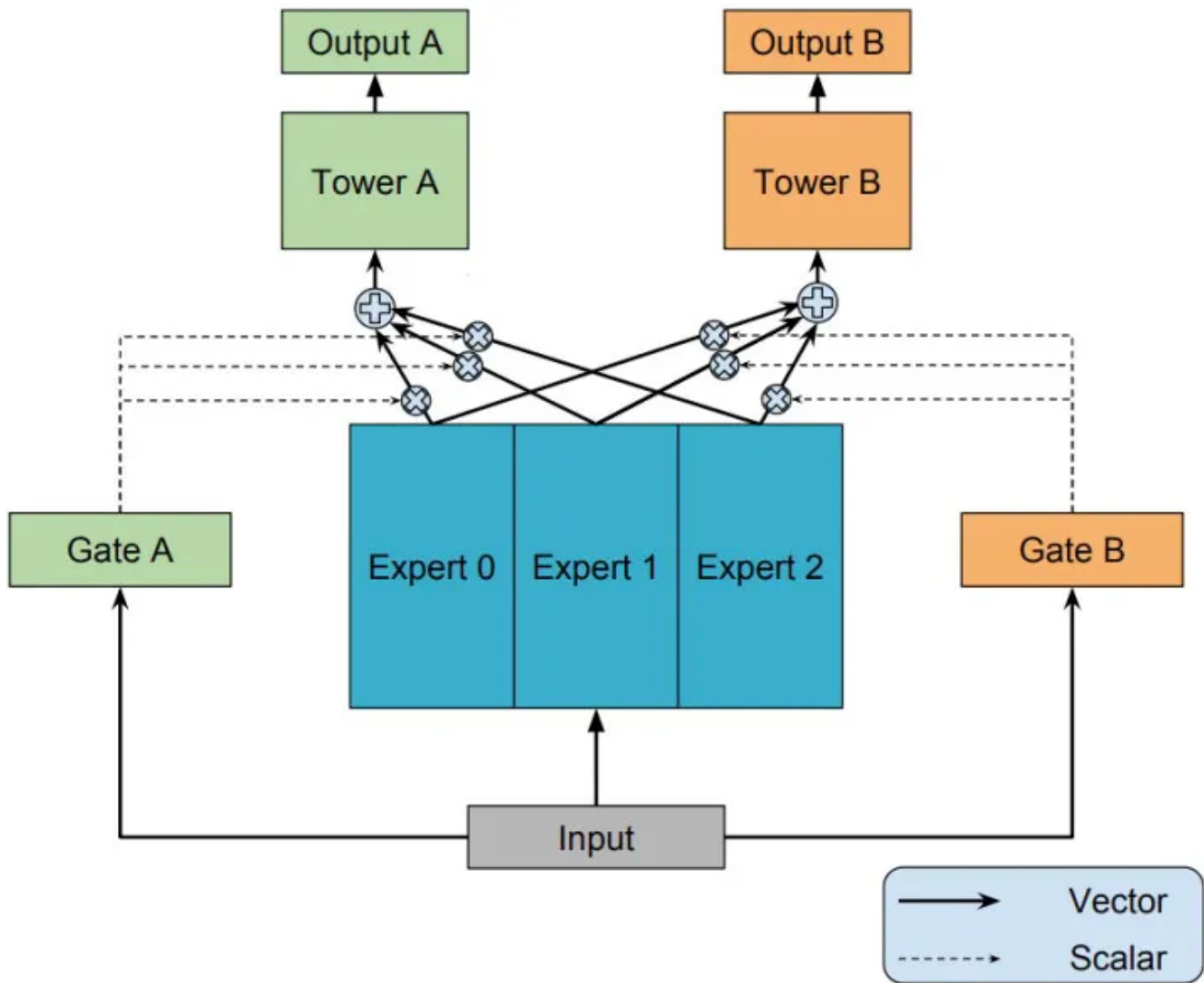
MoE 模型可以形式化表示为：

$$y = \sum_{i=1}^n g(x)_i f_i(x)$$

其中， $f_i (i = 1, \dots, n)$ 是 n 个 expert network（expert network 可认为是一个神经网络）， g 是组合 experts 结果的门控网络（gating network）， $\sum_{i=1}^n g(x)_i = 1$ ，具体来说 g 产生 n 个 experts 上的概率分布，最终的输出是所有 experts 的带权加和。显然，MoE 可看做基于多个独立模型的集成方法。这里注意 MoE 只对应上图中的一部分，我们把得到的带权结果 y 输入到子任务分别对应的 tower network 中进行学习。上文中也提到了有些文章将 MoE 作为一个基本的组成单元，将多个 MoE 结构堆叠在一个大网络中。比如一个 MoE 层可以接受上一层 MoE 层的输出作为输入，其输出作为下一层的输入使用。

- Multi-gate MoE model

知道了 OMoE 结构，那么 MMoE（Multi-gate Mixture-of-Experts）的结构就很容易猜出来了，通过名字里的 Multi-gate 很容易想到，MMoE 就是在 OMoE 的基础上，用了多个门控网络，其核心思想是将 shared-bottom 网络中的函数 f 替换成 MoE 层，结构如下图所示：



MMoE 可以形式化表达为：

$$y_k = h^k(f^k(x)), f^k(x) = \sum_{i=1}^n g^k(x)_i f_i(x)$$

其中， g^k 是第 k 个任务中组合 experts 结果的门控网络（gating network），注意每一个任务都有一个独立的门控网络。 $g^k(x) = \text{softmax}(W_{gk}x)$ ，它的输入是 input feature，输出就是所有 Experts 上的权重。 $f_i (i = 1, \dots, n)$ 是 n 个专家（expert）网络。

一方面，因为gating networks通常是轻量级的，而且expert networks是所有任务共用，所以相对于论文中提到的一些baseline方法在计算量和参数量上具有优势。

另一方面，MMoE 其实是 MoE 针对多任务学习的变种和优化，相对于 OMoe 的结构中所有任务共享一个门控网络，MMoE 的结构优化为每个任务都单独使用一个门控网络。这样的改进可以针对不同任务得到不同的 Experts 权重，从而实现对 Experts 的选择性利用，不同任务对应的门控网络可以学习到不同的 Experts 组合模式，因此模型更容易捕捉到子任务间的相关性和差异性。

5.3.4 MMoE的应用

在2019年的RecSys上，Google又发表了一篇MMoE的应用文章《Recommending What Video to Watch Next: A Multitask Ranking System》。

一般推荐系统排序模块的进化路径是：CTR任务 -> CTR + 时长 -> MultiTask & Selection Bias。该论文主要聚焦于大规模视频推荐中的排序阶段，介绍一些比较实在的经验和教训，解决 **Multitask Learning, Selection Bias** 这两个排序系统的关键点。算法的应用离不开其场景，这篇文章应用在Youtube视频推荐的场景中，其优化的多目标包含：

- 参与度目标（engagement objectives）：点击率、完播率；
- 满意度目标（satisfaction objectives）：喜欢、评分；

有很多不同甚至是冲突的优化目标，比如我们不仅希望用户观看，还希望用户能给出高评价并分享。对于存在潜在冲突的多目标，通过 MMoE 的结构来解决，通过门结构来选择性的从输入获取信息。

在真实场景中的另一个挑战是系统中经常有一些隐性偏见，比如用户是因为视频排得靠前而点击 & 观看，而非用户确实很喜欢。因此用之前模型产生的数据会引发 bias，从而形成一个反馈循环，越来越偏。例如在Youtube详情页的场景大多都面临着各种bias问题。包括 position bias, user bias, trigger bias等。position bias指的是不同坑位天然的xx率指标都不同；user bias表示着不同用户天然的xx率不同，有的用户爱点，有的用户不爱点；trigger bias表示这个内容出现在不同的详情页下，xx率也并不相同。如何高效有力地解决这些偏差是个尚未解决的问题。

Youtube这篇论文解决的是position bias的问题，下图以点击率为例，可以看出Youtube的详情页场景的点击率面临着严重的position bias问题。



"Large-Scale Deep Learning with TensorFlow," Jeff Dean

18,670 views

228

7

SHARE

SAVE

...



Association for Computing Machinery (ACM)

Published on Aug 9, 2016

Title: Large-Scale Deep Learning with TensorFlow

Date: Thursday, July 07, 2016

SHOW MORE

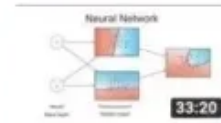
SUBSCRIBE 16K

Up next

AUTOPLAY



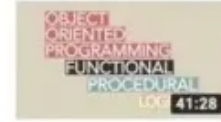
How Will Artificial Intelligence Affect Your...
TEDx Talks
348K views



A friendly introduction to Deep Learning and Neur...
Luis Serrano
419K views



Why graphene hasn't taken over the world...yet
Verge Science
10M views



4 Programming Paradigms In 40 Minutes
Coding Tech
151K views

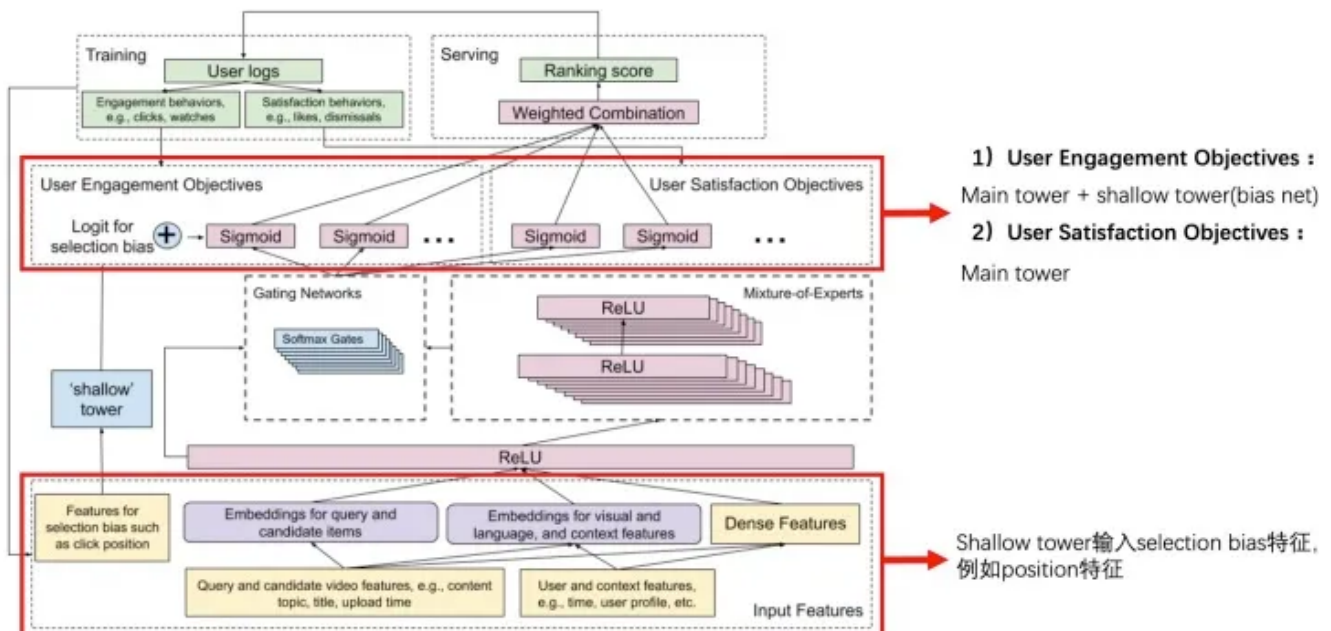


Tensorflow and deep learning - without a PhD...
Devbox
480K views



Heroes of Deep Learning: Andrew Ng interviews Ia...
Preserve Knowledge
77K views

解决这个问题的方式是使用一个shallow tower来建模点击率的position bias。这个shallow tower的输入特征只采用一些selection bias特征，结果加入到user engagement objectives中（Youtube认为这些指标的position bias比较大）。



6. MMoE+ESMM结合

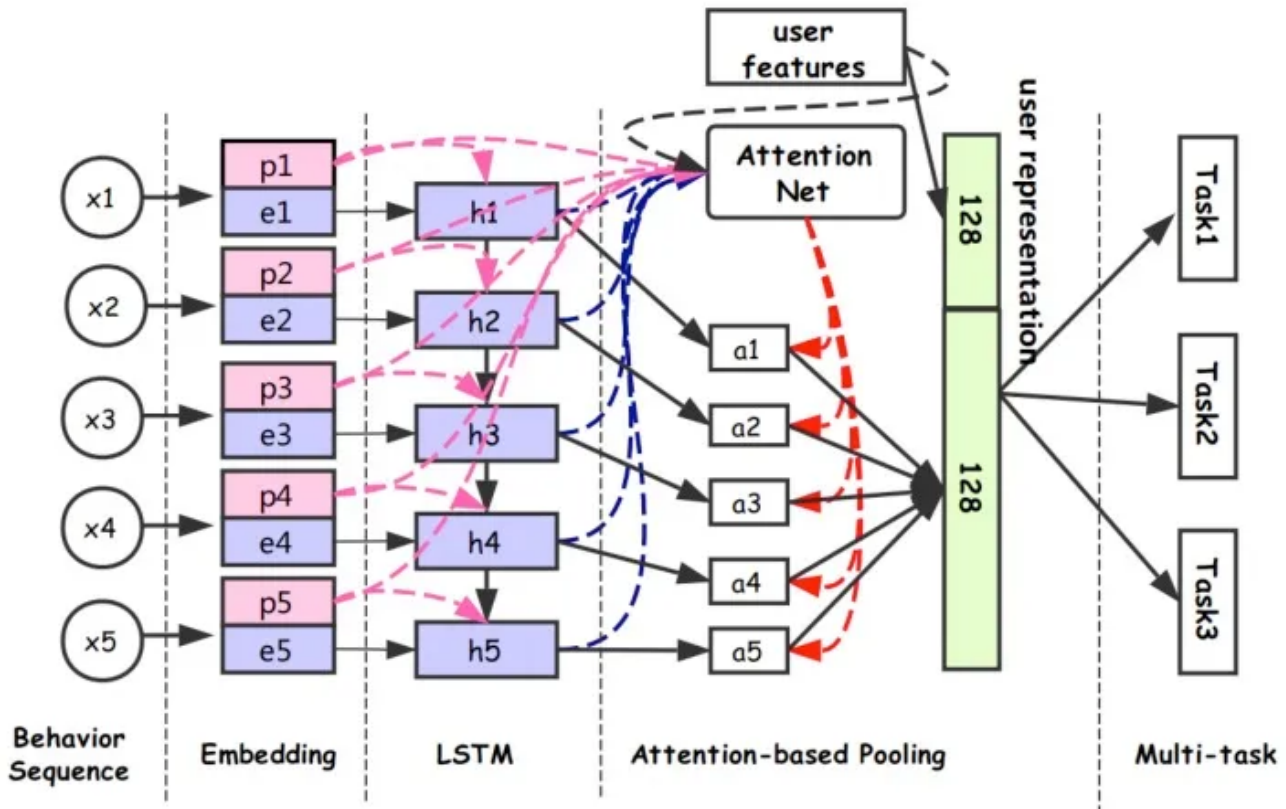
MMoE是使用expert+gate的方式替代hard parameter sharing，让不相关的任务也可以学习的很好。ESMM聚焦于寻找多目标的任务中本身就存在的联系，来解决SSB和DS等问题，更好的建模。那么这两类方法能不能结合呢？是我们后续需要探索的方向。

7. 其它关于MTL模型在推荐场景的工作

7.1 阿里 DUPN

对应论文：《Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-commerce Tasks》

多任务学习的优势：可共享一部分网络结构，比如多个任务共享embedding参数。学习到的用户、商品embedding向量可迁移到其它任务中。本文提出了一种多任务模型DUPN：

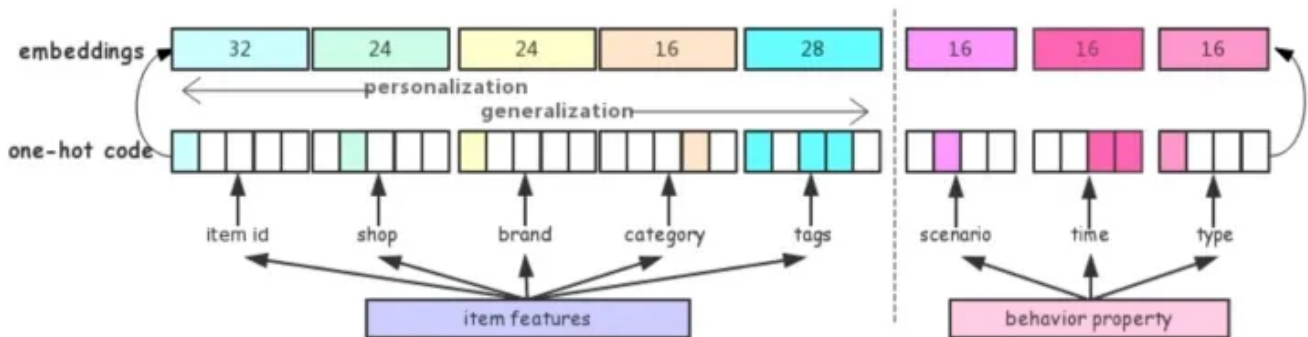


DUPN模型分为行为序列层、Embedding层、LSTM层、Attention层、下游多任务层。

(1) 行为序列层：输入用户的行为序列 $x = \{x_1, x, \dots, x_N\}$ ，其中每个行为都有两部分组成，分别是item和property项。**property**项表示此次行为的属性，比如场景（搜索、推荐等场景）、时间、类型（点击、购买、加购等）。**item**包括商品id和一些side-information(比如店铺id、brand等)。其实很多场景都用了side-information，我理解主要原因有两点：

- **side-information**的embedding量级要比商品id小很多，这样用**side-information**来近似替代**item_id**，训练起来更容易；
- 可一定程度上解决商品的冷启动问题，因为虽然很多时候拿不到**item id**，但是还是能获得一些属性信息的。

(2) Embedding层，主要多item和property的特征做处理。



(3) LSTM层：得到每一个行为的Embedding表示之后，首先通过一个LSTM层，把序列信息考虑进来。

(4) Attention层：区分不同用户行为的重要程度，经过attention层得到128维向量，拼接上128维的用户向量，最终得到一个256维向量作为用户的表达。

(5) 下游多任务层：CTR、L2R(Learning to Rank)、用户达人偏好FIFP、用户购买力度量PPP等。

另外，文中也提到了两点多任务模型的使用技巧：

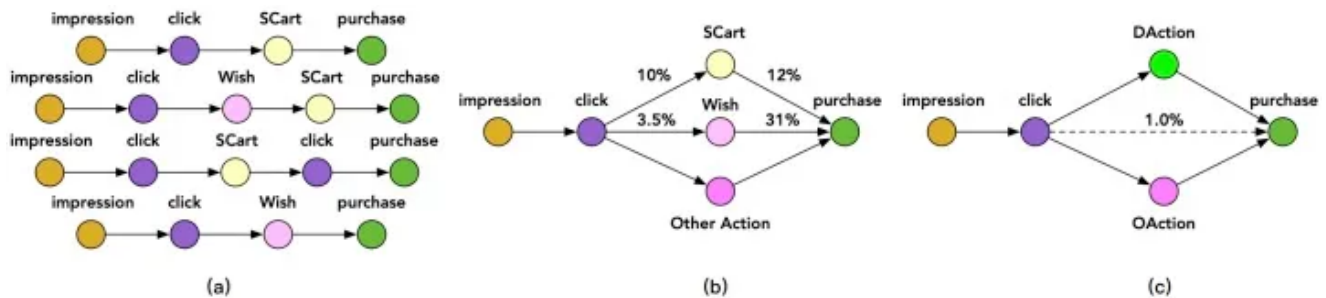
- 天级更新模型：随着时间和用户兴趣的变化，ID特征的Embedding需要不断更新，但每次都全量训练模型的话，需要耗费很长的时间。通常的做法是每天使用前一天的数据做增量学习，这样一方面能使训练时间大幅下降；另一方面可以让模型更贴近近期数据。
- 模型拆分：由于CTR任务是point-wise的，如果有1w个物品的话，需要计算1w次结果，如果每次都调用整个模型的话，其耗费是十分巨大的。其实User Representation只需要计算一次就好。因此我们会将模型进行一个拆解，使得红色部分只计算一次，而蓝色部分可以反复调用红色部分的结果进行多次计算。

7.2 阿里ESM²

对应论文：之前论文名字《Conversion Rate Prediction via Post-Click Behaviour Modeling》，现在论文名字《Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction》，对应的都是一篇论文。

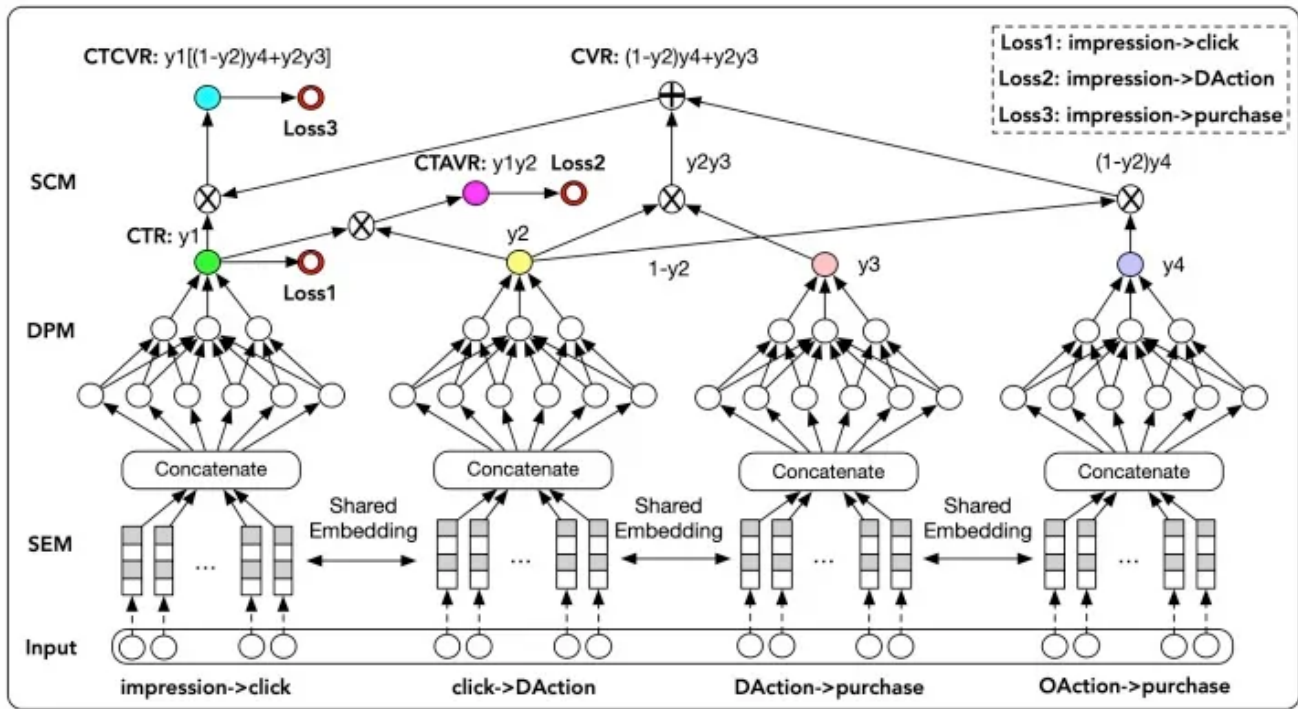
ESM² 可以看作是ESMM的升级版。ESMM中有两个子网络，分别是Main Task用于预估CVR值，Auxiliary Tasks用于预估CTR值。两个网络共享Embedding部分。Loss分为两部分，一是CTR预估带来的loss，二是pCTCVR（pCTR * pCVR）带来的loss。CTCVR是从impression到buy，CTR是从impression到click，所以CTR和CTCVR都可以从整个impression样本空间进行训练，一定程度的消除了样本选择偏差。

但对于CVR预估来说，ESMM模型仍面临一定的样本稀疏问题，因为click到buy的样本非常少。但其实一个用户在购买某个商品之前往往会有一些其他的行为，比如将商品加入购物车或者心愿单。如下图所示：



文中把加购物车或者心愿单的行为称作Deterministic Action (DAction)，表示购买目的很明确的一类行为。而其他对购买相关性不是很大的行为称作Other Action(OAction)。那原来的 Impression→Click→Buy购物过程就变为Impression→Click→DAction/OAction→Buy过程。

ESM² 模型结构：



那么该模型的多个任务分别是：**Y1**:点击率；**Y2**:点击到DAction的概率；**Y3**:DAction到购买的概率；**Y4**:OAction到购买的概率。

并且从上图看出，模型共有3个loss，计算过程分别是：

- **pCTR**: Impression→Click的概率是第一个网络的输出。
- **pCTAVR**: Impression→Click→DAction的概率， $pCTAVR = Y1 * Y2$ ，由前两个网络的输出结果相乘得到。
- **pCTCVR**: Impression→Click→DAction/OAction→Buy的概率， $pCTCVR = CTR * CVR = Y1 * [(1 - Y2) * Y4 + Y2 * Y3]$ ，由四个网络的输出共同得到。其中 $CVR = (1 - Y2) * Y4 + Y2 * Y3$ 。是因为从点击到DAction和点击到OAction是对立事件。

随后通过三个logloss分别计算三部分的损失，最终损失函数由三部分加权得到。具体公式我就不再贴了，可以去源论文看一下。

7.3 阿里 DeepMCP

对应论文：《Representation Learning-Assisted Click-Through Rate Prediction》

DeepMCP模型是阿里19年提出的一个广告点击率预估，不同于传统的CTR预估模型刻画特征-CTR之间的联系，该模型采用多任务学习的方式进行联合训练进一步挖掘用户-广告、广

告-广告之间的信息从而使得系统对于特征-CTR之间联系的刻画更加准确。该模型主要包括 Prediction subnet、Matching subnet、Correlation subnet。模型结构如下：

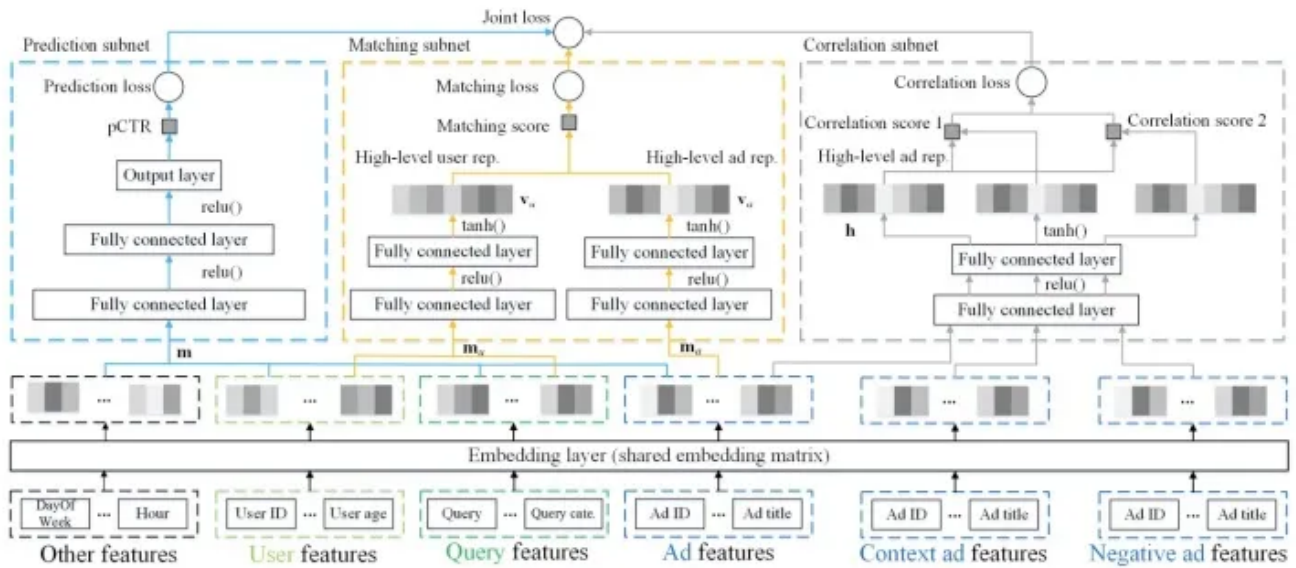


Figure 4: Detailed view of the DeepMCP model. The prediction, matching and correlation subnets share the same embedding matrix.

- Prediction subnet

这一部分可以换成任意CTR预估DNN结构，本文使用的是简单的MLP。

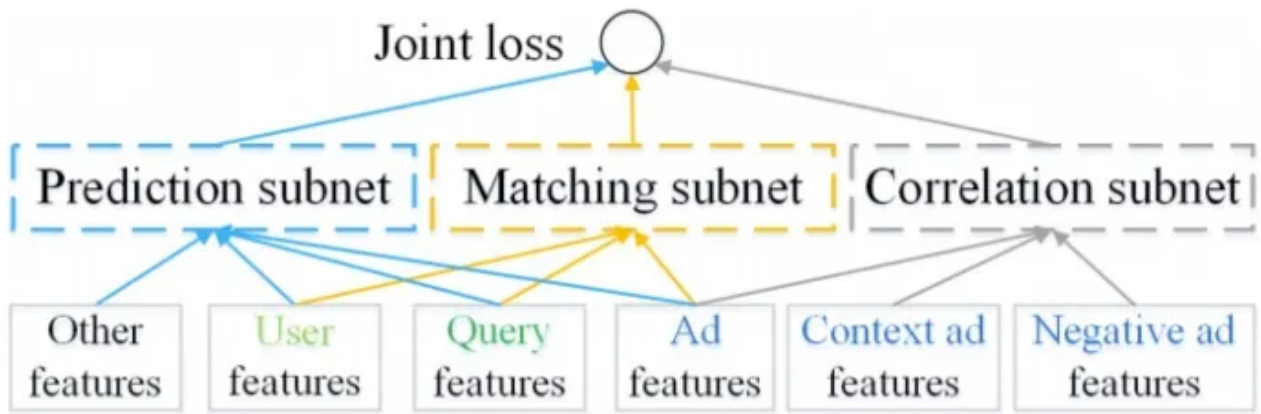
- Matching subnet

Matching subnet主要是捕捉user-item的关系，帮助学习到更有用的user和item的表示。推荐领域中，传统矩阵分解方法，通常是通过user id以及item id的隐向量的内积得到rating score。而本文使用的是user以及item的全部feature。Matching subnet是一个双塔结构，将user feature以及query feature输入其中一侧得到高阶user表示，另一部分则是item feature得到的表示。值得注意的是，这里最后一层得到的表示，经过的激活函数是tanh而非ReLU，原因是ReLU会导致大量维度为0，从而导致向量点积接近0。本文实验使用的loss是point-wise的loss，作者说明了同样可以换成pair-wise。

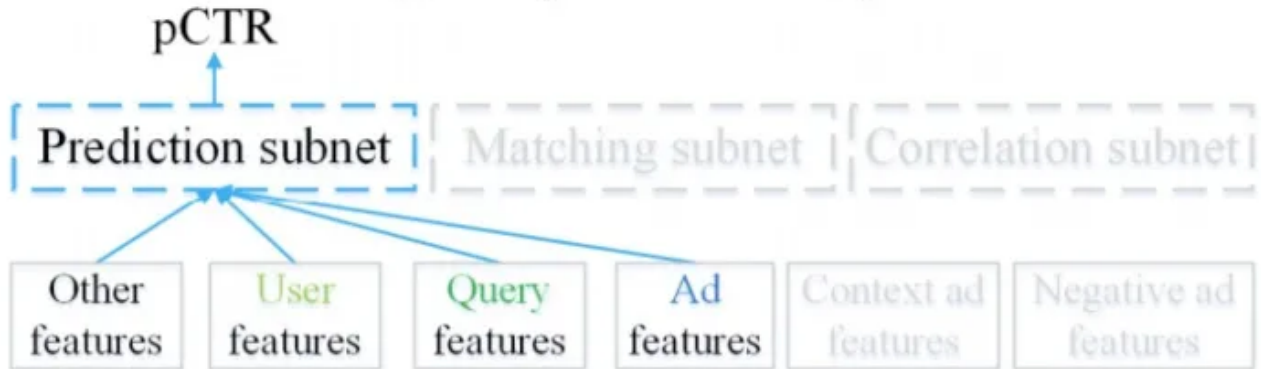
- Correlation subnet

这一部分主要是刻画item-item关系。这里使用skip gram形式，同一个时间窗内用户点击序列认为他们之间存在关系，作为正样本，不在的则为负样本，学习表示。

训练及预测如下图：



(a) DeepMCP - Training



(b) DeepMCP - Testing

训练过程三部分联合训练，共享embedding部分参数。线上测试部分只需要Prediction subnet，得到结果。

7.4 NMTR

对应论文：《Neural Multi-task Recommendation from Multi-behavior Data》

大多数现有的推荐系统只会用到用户单一类型的行为数据，例如在电子商务中可能只会使用到用户购买的行为数据。但是其他类型的用户行为数据也可以提供非常有价值的信号，例如点击、加购物车等。对此提出了一种新的解决方案NMTR（Neural Multi-Task Recommendation）来学习用户的多行为数据。

在现有的诸多模型中，大家都尝试挖掘和利用用户的其他行为数据，例如ESMM等，用到了用户的点击行为来辅助CVR等任务的学习。这篇文章所述的方法在很多实践中也都有所验证，例如Zohar Komarovsky的这篇博客《Deep Multi-Task Learning — 3 Lessons Learned》中也谈到了类似的实践经验。地址：<https://towardsdatascience.com/deep-multi-task-learning-3-lessons-learned-7d0193d71fd6>

该论文的核心思想就是：

- 采用MTL的方式从用户的多行为数据中进行建模；
- 为了捕捉用户的行为关系，采用级联(Cascaded way)的方式进行构建；

NMTR模型的核心框架如下：

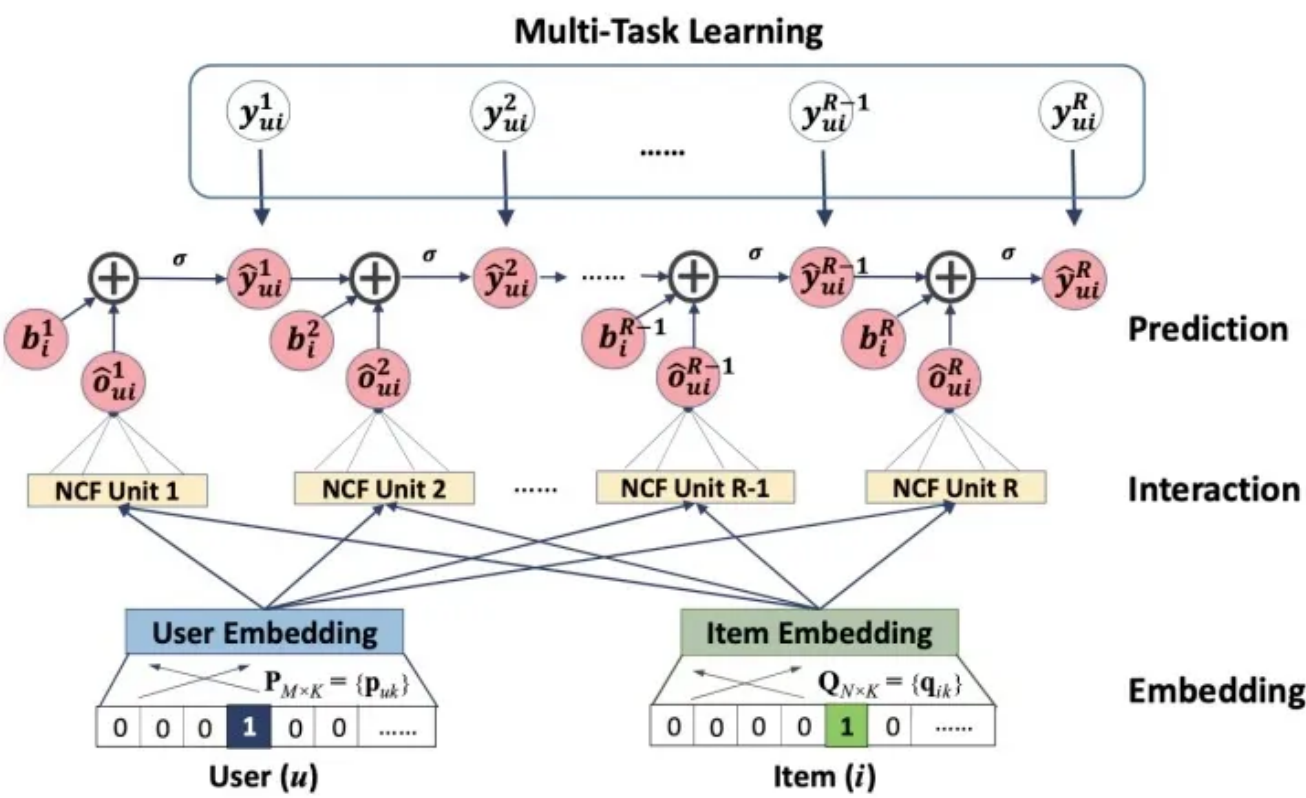


Fig. 1. Illustration of our proposed NMTR model.

8. 多目标预估中的其它问题

通过多任务学习训练一个模型预估多个目标，然后线上融合多个目标进行排序。多个目标融合的时候很多公司都是加权融合，比如更看重时长可能时长的权重就大些，更看重分享，分享的权重就大些，加权系数一般通过AB实验调整然后固定，这样带来的问题就是，当模型不断迭代的时候，这个系数可能就不合适了，经常会出现的问题是加权系数影响模型的迭代效率。具体多个目标怎么融合，这里面机制发挥的空间比较大，当然也有一些这方面的研究工作。

（1）最简单的办法，我们可以整合不同tasks的loss function，然后简单求和。这种方法存在一些不足，比如当模型收敛时，有一些task的表现比较好，而另外一些task的表现却惨不

忍睹。其背后的原因是不同的损失函数具有不同的尺度，某些损失函数的尺度较大，从而影响了尺度较小的损失函数发挥作用。这个问题的解决方案是把多任务损失函数“简单求和”替换为“加权求和”。加权可以使得每个损失函数的尺度一致，但也带来了新的问题：加权的超参难以确定。幸运的是，有一篇论文《Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics》通过“不确定性(uncertainty)”来调整损失函数中的加权超参，使得每个任务中的损失函数具有相似的尺度。该算法的keras版本实现，详见GitHub地址：[yaringal/multi-task-learning-example](https://github.com/yaringal/multi-task-learning-example)

(2) 最近基于贝叶斯优化的自动调参方法对于多目标参数寻优是一个很火热的讨论方向。

(3) 还有多目标常常存在的问题是，指标a的提升常常伴随着指标b的下降。阿里在recsys 2019发表的文章《A Pareto-Efficient Algorithm for Multiple Objective Optimization in E-Commerce Recommendation》中使用帕累托最优的思想，生成多组权重参数，目的是为了在不损害a指标的情况，提升b指标，感兴趣的同学也可以看一下。

9. 工业界案例分享

多目标排序和多任务学习的应用，这两年在工业界有很多尝试，下面是一些经典案例：

(1) 美团

- 美团“猜你喜欢”深度学习排序模型实践，地址：<https://mp.weixin.qq.com/s/jdRucishwV8qBmGLTFJCA>

(2) 知乎

- 进击的下一代推荐系统：多目标学习如何让知乎用户互动率提升100%？，地址：<https://mp.weixin.qq.com/s/J0j9NwSNhxab6bXqBBzaUw>
- 「回顾」知乎推荐页Ranking经验分享，地址：<https://mp.weixin.qq.com/s/GUMz-HfbjQvzdGVQkKz4zA>

(3) 美图

- 当推荐遇到社交：美图的推荐算法设计优化实践，地址：<https://mp.weixin.qq.com/s/Eih4J51C8Eh-cuZ8vznESg>

- 多任务学习在美图推荐排序的近期实践，地址：<https://mp.weixin.qq.com/s/-rw2Gecv-QOSW33Q8QTfcA>

(4) 阿里

- 阿里淘宝：Multi Task Learning在工业界如何更胜一筹，地址：<https://developer.aliyun.com/article/568166>
- 阿里UC：UC 信息流推荐模型在多目标和模型优化方面的进展，地址：<https://mp.weixin.qq.com/s/FXlxT6qSridawZDIdGD1mw>
- 阿里淘系：从谷歌到阿里，谈谈工业界推荐系统多目标预估的两种范式，地址：<https://zhuanlan.zhihu.com/p/125507748>
- 阿里UC：信息流短视频时长多目标优化(从技术角度聊聊，短视频为何让人停不下来？)，地址：<https://mp.weixin.qq.com/s/lb5b-7ImTI0hlFwIBkpqxQ>
- 阿里1688：深度学习在阿里B2B电商推荐系统中的实践，地址：https://mp.weixin.qq.com/s/OU_alEVLPyAKRjVIDv1o-w

(5) YouTube

- Youtube 排序系统：Recommending What Video to Watch Next，地址：<https://zhuanlan.zhihu.com/p/82584437>

(6) 蘑菇街

- 电商多目标优化小结，地址：<https://zhuanlan.zhihu.com/p/76413089>

(7) 腾讯

- 微信「看一看」推荐排序技术揭秘，地址：https://mp.weixin.qq.com/s/_hGIIdl9Y7hWuRDeCmZUMmg
- 详文解读微信「看一看」多模型内容策略与召回，地址：<https://mp.weixin.qq.com/s/s03njUVj1gHTOS0GSVgDLg>

(8) 花椒直播

- 深度学习在花椒直播中的应用——排序算法篇，地址：<https://mp.weixin.qq.com/s/e6Spp7smIEUUEXjxHzUOFA>

(9) 微博

- 机器学习在热门微博推荐系统的应用，地址：<https://mp.weixin.qq.com/s/9oguGl72WQqSVRJc7CRvEQ>

(10) 京东

- 【技术分享】京东电商广告和推荐的机器学习系统实践，地址：
https://mp.weixin.qq.com/s/PYMR97nIzS3yp5-vkyhM_w

(11) BIGO

- BIGO | 内容流多目标排序优化，地址：https://mp.weixin.qq.com/s/3AMW-vUr2S9FBSDUr_JhpA

10. Reference

由于参考的文献较多，我把每篇参考文献按照自己的学习思路，进行了详细的归类和标注。

推荐系统中多目标综述文章：

- 推荐系统中的多任务学习，地址：<https://lumingdong.cn/multi-task-learning-in-recommendation-system.html#dfref-footnote-5>
- 从谷歌到阿里，谈谈工业界推荐系统多目标预估的两种范式，地址：
<https://mp.weixin.qq.com/s/NCtTgEh8iRRZGhcrS6Gd8g>
- 推荐系统中的多目标学习 - 奔奔的文章 - 知乎
<https://zhuanlan.zhihu.com/p/183760759>
- 多任务学习在推荐算法中的应用，地址：<https://mp.weixin.qq.com/s/-SHLp26oGDDp9HG-23cetg>
- Multi-task多任务模型在推荐算法中应用总结1 - 梦想做个翟老师的文章 - 知乎
<https://zhuanlan.zhihu.com/p/78762586>
- Multi-task多任务学习在推荐算法中应用(2) - 梦想做个翟老师的文章 - 知乎
<https://zhuanlan.zhihu.com/p/91285359>
- 深度总结 | 多任务学习方法在推荐中的演变，地址：
https://mp.weixin.qq.com/s/ifTNRW0W7-P_LyfNldtavQ
- 推荐精排模型之多目标模型 - billlee的文章 - 知乎
<https://zhuanlan.zhihu.com/p/221738556>
- 推荐系统中如何做多目标优化，地址：
https://mp.weixin.qq.com/s/z7bLfTujt3tP035YgQv_jQ

- 推荐系统rank模块-多目标排序算法(一) - 邁書傑的文章 - 知乎
<https://zhuanlan.zhihu.com/p/65001130>

推荐多目标工业界案例：

- 进击的下一代推荐系统：多目标学习如何让知乎用户互动率提升100%？，地址：
<https://mp.weixin.qq.com/s/J0j9NwSNhxab6bXqBBzaUw>
- UC 信息流推荐模型在多目标和模型优化方面的进展，地址：
<https://mp.weixin.qq.com/s/FXlxT6qSridawZDIdGD1mw>
- 当推荐遇到社交：美图的推荐算法设计优化实践，地址：
<https://mp.weixin.qq.com/s/Eih4J51C8Eh-cuZ8vznESg>
- 多任务学习在美图推荐排序的近期实践，地址：<https://mp.weixin.qq.com/s/-rw2Gecv-QOSW33Q8QTfcA>
- Multi Task Learning在工业界如何更胜一筹，地址：
<https://developer.aliyun.com/article/568166>

如何对不同目标做加权：

- 为什么基于贝叶斯优化的自动调参没有大范围使用？ - 知乎
<https://www.zhihu.com/question/33711002>
- 帕累托的角度解推荐系统中的多目标优化问题，地址：
https://mp.weixin.qq.com/s/xqlvFRts3S29DAgzBV3_ca
- 推荐多目标进阶之自适应权重学习 - billlee的文章 - 知乎
<https://zhuanlan.zhihu.com/p/238125972>
- 多目标学习之自适应权重学习进阶 - billlee的文章 - 知乎
<https://zhuanlan.zhihu.com/p/252986597>

多任务模型学习：

- 详解谷歌之多任务学习模型MMoE(KDD 2018)，地址：
<https://zhuanlan.zhihu.com/p/55752344>
- 多目标优化概论及基础算法ESMM与MMOE对比，地址：
<https://mp.weixin.qq.com/s/8o2CEMH2XLmo6NwNsMX23w>
- Youtube 排序系统：Recommending What Video to Watch Next - 魔鬼吊儿郎的文章 - 知乎
<https://zhuanlan.zhihu.com/p/82584437>
- 多目标学习与推荐系统 - 雪的味道的文章 - 知乎
<https://zhuanlan.zhihu.com/p/144030768>
- 推荐系统多任务学习上分神技！，地址：
<https://mp.weixin.qq.com/s/8quc62wB4cXTcZaesmR7JQ>

- Deep Multi-Task Learning — 3 Lessons Learned，地址：
<https://towardsdatascience.com/deep-multi-task-learning-3-lessons-learned-7d0193d71fd6>
- Multi-task Learning的三个小知识 - mountain blue的文章 - 知乎
<https://zhuanlan.zhihu.com/p/56613537>



长按二维码扫描关注

Microstrong

ID:MicrostrongAI

Microstrong(小强)同学主要研究兴趣是机器学习、深度学习、推荐系统、自然语言处理等相关内容，分享在学习过程中的读书笔记！期待您的关注，欢迎一起学习交流进步！

收录于话题 #推荐系统原理解析与实践部署·8个

下一篇

喜欢此内容的人还喜欢

网易云音乐推荐中的用户行为序列深度建模

DataFunTalk

李宏毅强化学习完整笔记！开源项目《LeeDeepRL-Notes》发布

Datawhale

图解 Attention（完整版）！

Datawhale