第 11 章 粗排模型

在上一章节中，我们系统介绍了工业级推荐系统中的召回模型。召回模块位于整个推荐系统级联架构的最上游，其主要目标是在海量物品库中快速筛选出少量高质量候选，为后续粗排、精排以及重排模型提供输入。为了兼顾召回精度、覆盖率以及多样性，工业界通常采用协同过滤召回、图召回、Embedding 向量召回、序列召回、多兴趣召回等多路召回模型协同工作的方式，不同召回通道分别从不同角度刻画用户兴趣，共同构成推荐系统的候选集。

相比召回阶段强调多路召回策略，排序阶段通常围绕一个主模型不断进行迭代优化。粗排、精排以及重排虽然都属于排序阶段，但它们承担着不同的职责。其中，粗排模型负责从召回阶段返回的大规模候选集合中快速筛选出少量高质量候选，在保证较高召回率的同时尽可能提升候选集合的整体质量；精排模型则利用更加复杂的网络结构与更加丰富的特征，对候选进行精细排序；而重排模型则进一步结合多样性、公平性、业务约束以及长期价值等因素，对最终推荐结果进行整体优化。因此，粗排模型既承担着连接召回与精排的重要桥梁作用，也是整个推荐系统中计算效率与排序效果之间的重要平衡点。

由于粗排模型需要对数千甚至上万个候选进行实时打分，因此其计算效率通常受到严格限制。相比精排模型，粗排模型更加注重推理时延与吞吐能力，在模型设计上通常采用参数规模较小、网络结构更加轻量的模型，以满足工业级在线推荐系统对于低延迟、高并发的要求。同时，粗排模型也需要充分利用用户特征、物品特征以及上下文特征等多源信息，对候选进行有效筛选，尽可能保留真正具有潜在兴趣的物品，为后续精排模型提供高质量候选集合。因此，粗排模型既不能过于简单而导致大量优质候选被提前过滤，也不能过于复杂而影响整体推荐链路的在线时延，这也是工业界粗排模型设计过程中最重要的优化目标之一。

随着深度学习技术的发展，工业界粗排模型也经历了持续演进。早期粗排模型更多采用 Logistic Regression、GBDT 等传统机器学习模型，随后逐渐发展为以双塔模型、多塔模型以及轻量级 DNN 为代表的深度学习模型，并进一步引入知识蒸馏、模型压缩、多任务学习以及集合优化等技术，在有限计算资源下不断提升粗排效果。近年来，随着推荐系统规模不断扩大以及业务目标日益复杂，粗排模型不仅承担着多目标预估的任务，还逐渐开始承担候选集合优化以及跨阶段协同优化等更加复杂的职责，在整个推荐链路中的重要性也越来越高。

在本章节中，我们将系统介绍工业级推荐系统中粗排模型的发展脉络与工程实践。首先介绍双塔模型及其在粗排阶段的应用，然后进一步介绍多塔模型、轻量级 DNN 模型、粗排蒸馏模型以及粗排集合优化建模等典型方法。同时，我们也会结合工业界实际应用案例，从模型结构设计、特征工程、训练策略以及优化方法等多个方面展开分析，帮助读者全面理解工业级推荐系统中粗排模型的设计思想与工程实现。

需要说明的是，粗排模型在工业系统中的落地不仅涉及模型训练与排序策略，还包括高效的在线推理架构以及与上下游阶段的协同优化。对于经典粗排双塔结构的线上 Serving 架构，由于其涉及 Embedding 计算、向量检索、缓存优化以及在线系统部署等更偏工程基础设施的内容，将在后续第17章中进行详细介绍。同时，粗排模型与召回、精排等阶段之间的跨阶段协同优化问题，通常也被称为链路一致性优化，其核心目标是在不同排序阶段之间保持排序目标的一致性并减少阶段间的信息损失，相关内容将在后续第21章中展开讨论。

11.1 双塔模型

双塔模型（Two-Tower Model）是工业级推荐系统粗排阶段最经典的模型架构之一，其基本思想最早可以追溯到微软于 2013 年提出的 DSSM（Deep Structured Semantic Model）模型。此后，双塔模型不断发展，衍生出了 MV-DSSM、TDSSM 以及 DSSM 结构融合 FM、Transformer 等网络结构的多种变体，并逐渐成为搜索、广告以及推荐系统中最主流的语义匹配模型之一。双塔模型的核心思想如下：

定义 11.1 (双塔模型)

分别利用两个相互独立的神经网络对用户（User）和物品（Item）进行特征建模，将两侧输入映射到同一个低维语义空间中，并通过向量内积、余弦相似度等方式计算用户与物品之间的匹配程度。

与传统单塔模型直接学习用户与物品交叉特征不同，双塔模型能够提前离线计算物品 Embedding，仅在线计算用户 Embedding，再通过高效的向量相似度计算完成候选筛选，因此具有较高的计算效率和良好的可扩展性，尤其适用于需要实时处理海量候选物品的工业级粗排阶段。

随着工业推荐系统的发展，双塔模型不仅广泛应用于粗排模型，在 Embedding 召回、搜索排序以及广告检索等场景中也得到了大量应用。近年来，虽然出现了越来越多复杂的排序模型，但双塔架构由于其优秀的在线推理效率以及良好的工程扩展能力，仍然是工业界最重要的基础模型之一。

11.1.1 DSSM 模型

DSSM（Deep Structured Semantic Model，深度结构化语义模型）最早由微软研究院于 2013 年提出，发表于 CIKM 会议论文《Learning Deep Structured Semantic Models for Web Search using Clickthrough Data》[4]。该模型最初用于 Web 搜索中的 Query-Document 语义匹配，随后被广泛应用于推荐系统、广告排序以及信息检索等领域，并成为现代双塔模型的重要基础。

DSSM 模型的核心思想如下：

定义 11.2 (DSSM 模型)

DSSM 模型利用两个深度神经网络分别学习查询（Query）与文档（Document）的语义表示，并将两者映射到同一个低维语义空间中。在该语义空间内，语义相近的查询与文档具有更加接近的向量表示，而语义无关的样本则距离较远。将这一思想迁移到推荐系统中，Query 通常对应用户侧特征，Document 则对应物品侧特征，因此 DSSM 也成为推荐系统中双塔模型最经典的原型。

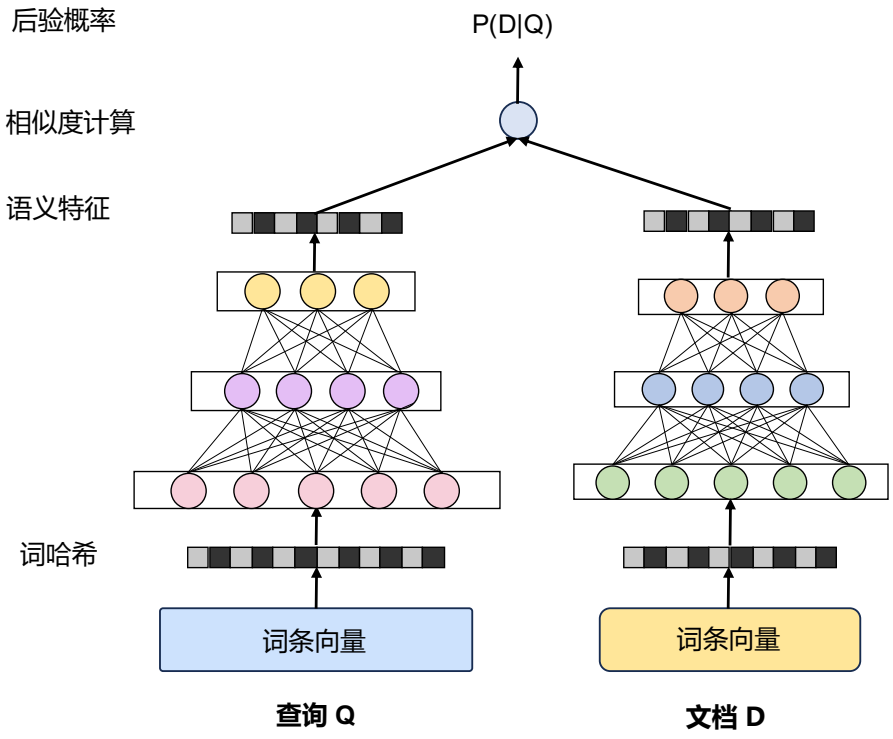

图 11.1: DSSM 模型整体结构示意图。

DSSM 模型整体结构如图 11.1所示。模型分别构建用户塔和物品塔，两侧网络通常采用多层全连接神经网络（MLP）作为编码器，对输入特征进行逐层非线性变换。设用户特征表示为 x_u，物品特征表示为 x_i，经过多层神经网络映射后分别得到用户 Embedding 和物品 Embedding：

$$\mathbf{e}_u = f_u(x_u), \qquad \mathbf{e}_i = f_i(x_i), \tag{11.1}$$

其中 $f_u(\cdot)$ 和 $f_i(\cdot)$ 分别表示用户塔与物品塔对应的深度神经网络。获得两侧 Embedding 之后，DSSM 采用余弦相似度作为用户与物品之间的匹配分数：

$$s(u,i) = \frac{\mathbf{e}_u^\top \mathbf{e}_i}{\|\mathbf{e}_u\|\|\mathbf{e}_i\|}. \tag{11.2}$$

相比直接采用内积计算，余弦相似度能够消除 Embedding 长度带来的影响，使模型更加关注语义方向上的一致性，因此也是 DSSM 论文中采用的匹配方式。

值得注意的是，DSSM 论文提出时，搜索系统需要处理几十万甚至数百万规模的词汇表。如果直接采用 One-Hot 表示作为神经网络输入，将导致输入维度过高、模型参数规模巨大，并且难以处理未登录词（Out-of-Vocabulary，OOV）问题。为此，论文提出了词哈希（Word Hashing）的技术，即利用 Letter Trigram 对单词进行编码。例如，对于单词“good”，首先在首尾加入特殊符号得到“#good#”，随后划分为“#go”、“goo”、“ood”以及“od#”等多个 Letter Trigram，并利用这些 Trigram 构成单词表示。相比传统 One-Hot 编码，Word Hashing 能够显著降低输入维度，同时天然具有较好的词形泛化能力，对于工业搜索系统中的超大规模词表具有重要意义。

在模型训练方面，DSSM 模型采用用户点击日志作为监督信号。对于每一个查询（Query），会同时构造一个点击文档以及若干随机采样得到的未点击文档，并首先根据余弦相似度计算查询与各候选文档之间的匹配得分，再利用 Softmax 函数得到点击文档的条件概率：

$$P(d^+|q) = \frac{\exp(\gamma s(q,d^+))}{\sum_{d\in D}\exp(\gamma s(q,d))}, \tag{11.3}$$

其中，γ 表示 Softmax 温度系数，D 表示候选文档集合，$s(q,d)$ 表示查询 q 与文档 d 之间的匹配得分。模型最终通过最大化点击文档的条件概率，即最小化交叉熵损失，对整个双塔网络进行端到端训练。

虽然 DSSM 模型最初面向 Web 搜索提出，但其所采用的双塔架构、共享语义空间以及基于点击日志进行监督学习的训练方式，对后续推荐系统的发展产生了深远的影响。如今工业界广泛应用的 YoutubeDNN、Embedding 召回、多塔粗排模型以及各种双塔模型（Two-Tower Model）的架构，本质上都继承了 DSSM 模型的基本思想，只是在输入特征、网络结构以及训练目标等方面进行了进一步扩展与优化。因此，DSSM 模型通常被认为是现代双塔模型以及工业级语义匹配模型的重要开端。

11.1.2 TDSSM 模型

TDSSM（Temporal Deep Structured Semantic Model）是微软于 2016 年提出的双塔模型，源自于 SIGIR 会议论文《Multi-Rate Deep Learning for Temporal Recommendation》[9]。该模型是在 DSSM 模型和 MV-DSSM 模型基础上的进一步扩展，其核心思想如下：

定义 11.3

TDSSM 模型是在用户表示学习过程中同时建模用户的长期兴趣（Long-term Preference）和短期兴趣（Temporal Preference），使模型既能够保持用户长期稳定的兴趣偏好，又能够快速捕获用户近期兴趣的动态变化，因此特别适用于新闻推荐、短视频推荐等具有较强时效性的推荐场景。

前面两个小节讲到的 DSSM 以及 MV-DSSM 模型，本质上都是利用用户历史行为构建一个静态（Static）的用户 Embedding。虽然这种表示能够较好地刻画用户长期兴趣，但它默认用户兴趣在一段时间内保持不变，因此难以及时响应用户兴趣的快速变化。例如，一位平时关注体育新闻的用户，在世界杯期间可能会集中浏览足球相关内容，而在旅游假期则可能更多关注酒店、美食等信息。如果仍然使用固定的用户 Embedding 进行推荐，模型往往无法及时捕获这些短期兴趣变化，从而降低推荐效果。

针对这一问题，TDSSM 首次将时间序列建模引入双塔模型，在保持长期兴趣建模能力的基础上，进一步增加了用户时序兴趣建模模块。整个模型结构如图 11.2所示。与传统 DSSM 模型相比，TDSSM 模型主要增加了用户时序兴趣网络（Temporal User Network），其中用户静态特征（User Static Features）和物品特征（Item Features）仍然采用多层全连接神经网络进行编码，而用户近期行为序列（User Temporal Features）则利用循环神经网络

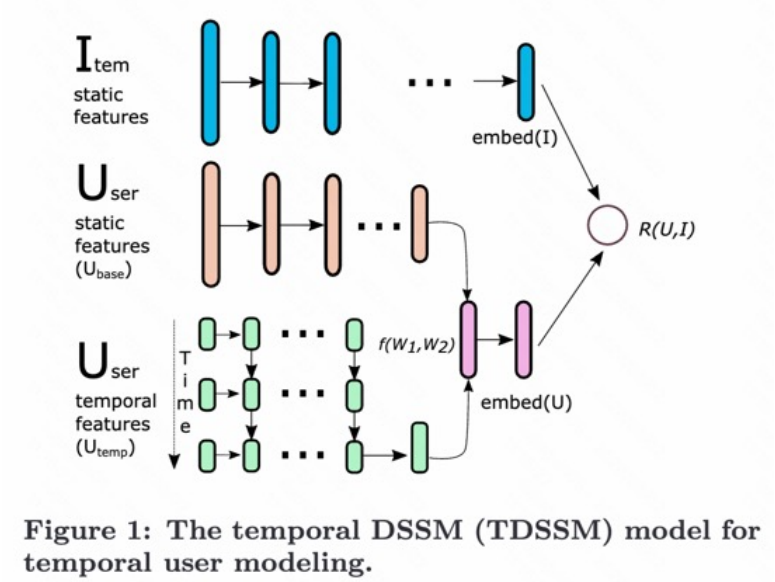

Figure 1: The temporal DSSM (TDSSM) model for temporal user modeling.

图 11.2: TDSSM 模型整体结构示意图。

（RNN）进行建模。TDSSM 模型的论文分别比较了 LSTM 与 GRU 两种网络结构，实验结果表明 LSTM 能够更稳定地建模用户兴趣随时间的演化，因此最终采用 LSTM 网络作为默认的序列编码器。

我们假设用户长期兴趣 Embedding 表示为

$$\mathbf{e}_{\text{base}} = f_{\text{base}}(x_u) \tag{11.4}$$

其中 x_u 表示用户静态特征，包括长期点击行为、兴趣标签以及用户画像等信息。用户近期行为序列经过 LSTM 编码后得到短期兴趣表示：

$$\mathbf{e}_{\text{temp}} = f_{\text{LSTM}}(S_u) \tag{11.5}$$

其中 S_u 表示按照时间顺序排列的用户行为序列。随后，将长期兴趣和短期兴趣进行融合，得到当前时刻用户最终的兴趣表示：

$$\mathbf{e}_u = f\left(\mathbf{e}_{\text{base}}, \mathbf{e}_{\text{temp}}\right) \tag{11.6}$$

其中 $f(\cdot)$ 表示兴趣融合函数。论文中主要讨论了三种融合方式：

- 元素级乘法（Element-wise Multiplication），即 $\mathbf{e}_u = \mathbf{e}_{\text{base}} \odot \mathbf{e}_{\text{temp}}$；
- 带权元素级乘法（Weighted Element-wise Multiplication），即 $\mathbf{e}_u = \mathbf{e}_{\text{base}} \odot \mathbf{w} \odot \mathbf{e}_{\text{temp}}$；
- 向量拼接（Concatenation），即 $\mathbf{e}_u = [\mathbf{e}_{\text{base}}; \mathbf{e}_{\text{temp}}]$。

根据上述融合方式融合之后的用户 Embedding 再与物品 Embedding 计算余弦相似度：

$$s(u, i) = \cos\left(\mathbf{e}_u, \mathbf{e}_i\right) \tag{11.7}$$

并通过 Softmax 计算点击概率，最终利用点击样本与随机负采样样本进行监督训练，其整体训练方式与 DSSM 模型保持一致。

相比传统 DSSM 模型，TDSSM 模型最大的特点在于：**将长期兴趣与短期兴趣进行了解耦建模**。长期兴趣主要反映用户较为稳定的偏好，例如长期关注科技、体育或财经等内容；而短期兴趣则描述用户当前一段时间内的即时需求，例如近期关注某部电影上映、世界杯赛事或者旅游攻略等。二者共同组成最终的用户 Embedding，使模型既具有良好的稳定性，又能够快速适应用户兴趣变化。

除了单一时间粒度建模之外，论文进一步提出了 Multi-Rate TDSSM（MR-TDSSM）模型，用于解决不同时间尺度下兴趣变化速度不同的问题。作者指出，如果时间窗口划分过细（例如按小时建模），虽然能够更准确地捕获用户兴趣变化，但行为数据十分稀疏，模型训练困难；而如果时间窗口划分过粗（例如按月建模），虽然数

据更加稳定，却难以及时响应用户兴趣变化。

因此，MR-TDSSM 模型采用多个不同时间尺度同时建模用户兴趣。例如，可以按照天、周以及两周等不同时间窗口分别构建多个用户行为序列，每个时间尺度对应一个独立的 LSTM 网络。其中，较短时间窗口对应的 Fast-RNN 主要负责建模用户近期兴趣变化，而较长时间窗口对应的 Slow-RNN 则用于学习用户中长期兴趣迁移。多个时间尺度学习得到的 Embedding 最终通过全连接网络进行融合，从而实现不同时间粒度行为信息的统一建模。这种设计能够兼顾兴趣变化的敏感性与模型稳定性，因此被称为 Multi-Rate TDSSM 模型。

由于 MR-TDSSM 模型需要同时训练多个 LSTM 网络，模型参数规模显著增加，训练成本也随之提高。为了提升训练效率，论文进一步提出了预训练的初始化策略。具体而言，首先利用前面介绍的 DSSM 模型对用户和物品进行预训练，学习得到较高质量的用户 Embedding 和物品 Embedding；随后，将这些 Embedding 作为 MR-TDSSM 模型的输入，而不是直接使用原始高维稀疏特征。这样不仅能够大幅降低输入维度和模型参数数量，还避免了重复训练物品侧深层网络，从而显著加快模型收敛速度，提高整体训练效率。

TDSSM 模型首次将时序兴趣建模引入双塔模型，使双塔模型不仅能够学习用户长期稳定的兴趣偏好，同时还能动态刻画用户兴趣随时间的变化规律。论文提出的长期兴趣与短期兴趣融合、多时间尺度兴趣建模以及预训练初始化等思想，对后续大量序列推荐模型以及双塔推荐模型产生了重要影响。虽然近年来 Transformer 逐渐取代 RNN 成为序列建模的主流方案，但 TDSSM 提出的静态兴趣与动态兴趣相结合的建模思想，至今仍然广泛应用于工业推荐系统中的用户兴趣建模任务。

这里需要强调的是，TDSSM 模型核心思想是在双塔结构中在用户侧网络引入用户兴趣建模，结合用户的短期兴趣与长期兴趣。虽然整个方法提出时间较早，在当前来看显得有些过时。但在工业级推荐系统的粗排模型中结合用户长短期兴趣建模仍然是粗排模型效果提升非常主要的抓手。在工业级推荐系统粗排模型 baseline 并不算太高的情况下，在粗排双塔模型结构中引入用户长短期兴趣建模通常能够带来显著的离线性能提升，也能在线上取得较大的 A/B 指标收益。

11.1.3 DSSM + Transformer 模型

近年来，随着 Transformer 在序列建模任务中的广泛应用，工业级推荐系统逐渐将 Transformer 引入双塔模型的用户塔（User Tower）中，用于增强用户兴趣表示能力。目前，无论是短视频、直播还是电商推荐场景，Transformer 已经成为工业级粗排模型用户侧建模的重要组成部分。

需要指出的是，Transformer 在工业界通常并不是作为独立的辅助网络存在，而是作为用户塔内部的一种特征编码器（Feature Encoder）。用户塔除了传统的用户画像特征、统计特征以及上下文特征之外，还会增加一个或多个 Transformer 模块，对用户行为序列进行建模，并将 Transformer 学习得到的序列表示与其他用户特征共同融合，最终生成用户 Embedding。因此，整个模型本质上仍然属于双塔模型，只是用户塔内部已经由单纯的 MLP 逐渐演变为由多个兴趣编码模块组成的复杂编码器。

DSSM+Transformer 模型整体结构如图 11.3所示。工业界用户兴趣通常可以划分为长期兴趣（Long-term Interest）和短期兴趣（Short-term Interest）两部分，因此用户塔内部通常也包含两个不同的 Transformer 编码模块。首先，短期兴趣主要用于刻画用户最近的兴趣变化。通常会选择用户最近几十个有效交互行为作为行为序列，例如最近 50 个点击或播放行为，组成用户 Action List：

$$\mathcal{S}_{short} = \{a_1, a_2, \cdots, a_T\} \tag{11.8}$$

其中 T 通常取 30∼100，在工业实践中最常见的配置为 50。每个行为通常包含多个离散特征与连续特征，例如视频 ID（Aid）、作者 ID（Pid）、观看时长（Watch Time）、播放完成率（Watch Time / Duration）、发生时间（Timestamp）、视频 Tag 等。这些特征首先映射为 Embedding，再对 Embedding 进行拼接（Concat）操作之后输入 Transformer 进行序列建模：

$$\mathbf{e}_{short} = \text{Transformer}(\mathcal{S}_{short}) \tag{11.9}$$

最终得到用户短期兴趣表示。另一方面，仅依赖最近几十个行为往往难以覆盖用户长期稳定的兴趣偏好。因此，近年来工业界越来越多地引入基于 SIM（Search-based Interest Model）[8] 等长序列建模方法，对用户数千甚至数

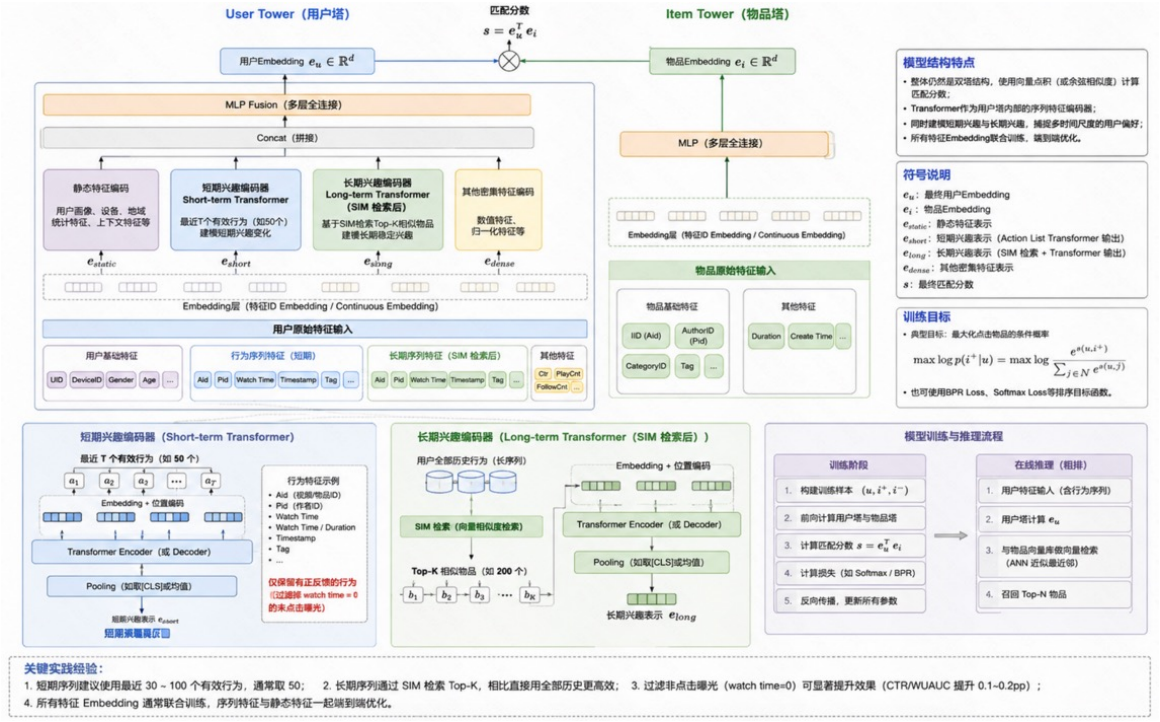

图 11.3: DSSM+Transformer 模型整体结构示意图。

万条历史行为进行兴趣检索。具体而言，首先利用 SIM 从长期行为库中检索与当前请求最相关的 Top-K 个历史物品：

$$\mathcal{S}_{long} = \text{TopK}(\text{SIM}(\mathcal{H})) \tag{11.10}$$

其中 $\mathcal{H}$ 表示用户完整历史行为序列。随后，再利用另一组 Transformer 对检索得到的 Top-K 兴趣序列进行编码：

$$\mathbf{e}_{long} = \text{Transformer}(\mathcal{S}_{long}) \tag{11.11}$$

从而获得用户长期兴趣表示。相比直接利用全部历史序列进行 Transformer 建模，这种" 先检索、后建模" 的方式能够显著降低 Transformer 的计算复杂度，同时保留与当前请求最相关的长期兴趣信息，因此已经成为工业界长序列兴趣建模的主流方案之一。

除了上述两类序列表示之外，用户塔还包含传统用户画像特征、统计特征以及上下文特征等静态信息，共同组成用户侧输入：

$$\mathbf{e}_{user} = \text{MLP}(\text{Concat}(\mathbf{e}_{static}, \mathbf{e}_{short}, \mathbf{e}_{long})) \tag{11.12}$$

其中 $\mathbf{e}_{static}$ 表示由用户画像、设备信息、地域信息等静态特征学习得到的 Embedding。Transformer 学习得到的长期兴趣 Embedding 与短期兴趣 Embedding 会直接拼接到用户塔底层特征中，再经过若干层 MLP 完成特征融合，最终得到用户 Embedding。随后，用户 Embedding 与物品塔输出的 Item Embedding 进行向量匹配：

$$s = \mathbf{e}_{user}^{\top} \mathbf{e}_{item} \tag{11.13}$$

得到最终的粗排匹配分数。

在工业界的粗排模型具体实践中，短期行为序列通常不会直接使用完整的曝光序列，而是会过滤掉未产生有效反馈的曝光行为，仅保留点击、播放等正反馈事件作为 Transformer 输入。这是因为大量未点击曝光通常包含较多噪声，容易干扰序列兴趣建模。实践经验表明，只使用有效的行为来构建用户行为序列（Action List）。相比起直接使用全部曝光序列，对 Action List 进行预处理去除噪声之后往往能够取得更加稳定的模型效果。

总的来说，DSSM+Transformer 模型仍然保持了双塔模型高效匹配的整体框架，但用户塔已经由传统的静态 MLP 编码器逐渐演变为融合静态特征、短期兴趣以及长期兴趣的多模块兴趣编码器。Transformer 负责学习不同时间尺度下的用户兴趣表示，而最终的用户 Embedding 仍然由整个用户塔共同生成。这种设计兼顾了双塔模型

高效在线推理能力与 Transformer 优秀的序列建模能力，因此已经成为当前工业级推荐系统粗排模型最主流的网络结构之一。

11.1.4 DAT 模型

DAT 模型源自美团发表于 DLP-KDD 2021 的论文《A Dual Augmented Two-tower Model for Online Large-scale Recommendation》[15]。传统双塔模型通过将用户与物品分别编码为独立的 Embedding 向量，并利用向量内积计算匹配分数，从而能够借助 Embedding 预计算和向量索引实现大规模候选集上的高效检索。然而，这种结构也带来了一个较为明显的问题：用户塔在编码用户时无法直接感知候选物品的信息，物品塔同样无法感知当前用户的信息，两个塔之间缺少充分的信息交互，因此模型的表达能力受到一定限制。

为了解决传统双塔模型中用户侧与物品侧信息相互独立的问题，DAT（Dual Augmented Two-tower）模型的核心思想如下：

定义 11.4

DAT 模型在传统双塔模型的基础上引入双向的增强向量（Augmented Vector），利用自适应模仿机制将用户和物品历史交互信息压缩到对应的增强向量中，并将其作为另一侧塔的额外输入，从而在不破坏双塔模型高效向量匹配结构的前提下，实现用户塔与物品塔之间的隐式信息交互；同时，通过类别对齐损失函数缓解不同类别数据分布不均衡导致的表示能力差异。

11.1.4.1 双塔增强结构

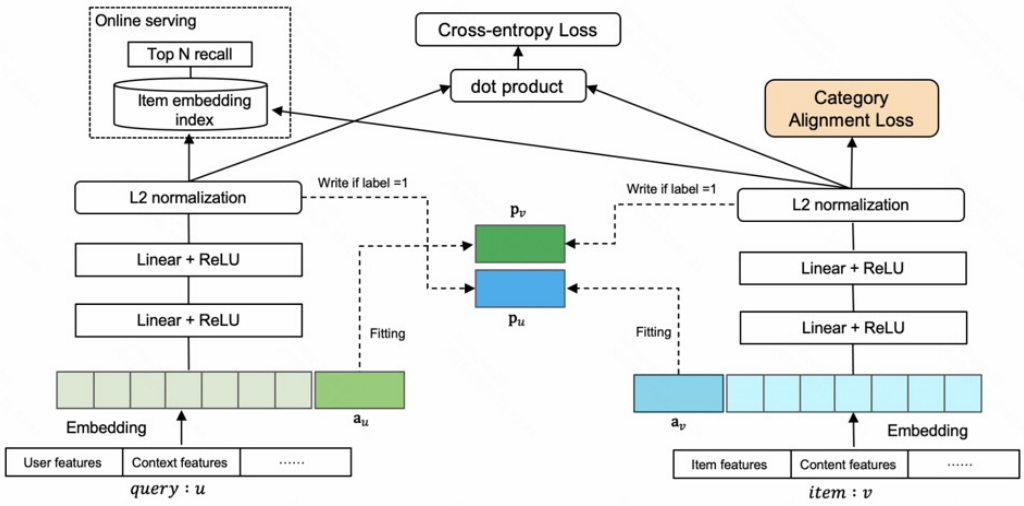

图 11.4: DAT 模型整体结构示意图。

DAT 模型的整体结构如图 11.4所示。DAT 模型首先保留传统双塔模型的基本结构。假设用户（Query）特征表示为 $\mathbf{u}$，物品特征表示为 $\mathbf{v}$，其中可以包含用户 ID、物品 ID、城市、性别、类别、价格以及其他业务特征。每个离散特征首先通过 Embedding 层映射为低维稠密向量，然后分别输入用户塔和物品塔进行特征编码。与传统双塔模型不同的是，DAT 模型并不是仅仅将当前用户自身的特征输入用户塔，也不是仅仅将当前物品自身的特征输入物品塔，而是分别为用户塔和物品塔引入一个额外的增强向量 $\mathbf{a}_u$ 和 $\mathbf{a}_v$。其中，$\mathbf{a}_u$ 用于描述用户历史上可能匹配的物品信息，而 $\mathbf{a}_v$ 则用于描述物品历史上可能匹配的用户信息。这样，用户塔和物品塔虽然仍然保持结构上的独立，但每个塔都可以通过增强向量获得另一侧的高层语义信息。

具体来说，假设用户侧原始特征经过 Embedding 后得到 $\mathbf{e}_u$，物品侧原始特征经过 Embedding 后得到 $\mathbf{e}_v$，则 DAT 模型分别构造增强后的输入表示：

$$\mathbf{z}_u = [\mathbf{e}_u, \mathbf{a}_u], \quad \mathbf{z}_v = [\mathbf{e}_v, \mathbf{a}_v] \tag{11.14}$$

其中，$\mathbf{a}_u$ 和 $\mathbf{a}_v$ 并不是当前用户与当前物品之间直接计算得到的交叉特征，而是通过历史正向交互学习得到的高层语义摘要。随后，增强后的输入分别经过多层全连接网络进行非线性变换。对于任意一侧塔，其第 l 层的隐藏表示可以表示为：

$$\mathbf{h}^{(l)} = \mathrm{ReLU}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \tag{11.15}$$

最终通过 L2 归一化得到用户 Embedding 和物品 Embedding，即：

$$\mathbf{p}_u = \mathrm{L2Norm}(\mathbf{h}_u^{(L)}), \quad \mathbf{p}_v = \mathrm{L2Norm}(\mathbf{h}_v^{(L)}) \tag{11.16}$$

在得到最终的用户 Embedding 和物品 Embedding 之后，DAT 模型仍然采用传统双塔模型的内积作为最终匹配分数，即：

$$s(u, v) = \langle \mathbf{p}_u, \mathbf{p}_v \rangle \tag{11.17}$$

因此，从在线服务的角度来看，如果将 DAT 模型应用在召回环节，DAT 模型仍然可以预计算物品 Embedding 并构建向量索引，并通过用户 Embedding 进行大规模 ANN 检索，并不会因为引入增强机制而完全失去双塔模型的高效检索能力。如果将 DAT 模型应用在粗排环节，DAT 模型仍然可以仍然可以预计算物品 Embedding，然后通过用户 Embedding 与物品 Embedding 进行向量内积计算匹配分数，并不会因为引入增强机制而显著增加在线推理时延。

11.1.4.2 自适应模仿机制

DAT 模型最核心的设计是自适应模仿机制（Adaptive-Mimic Mechanism，AMM）。AMM 的基本思想如下：

定义 11.5

AMM 希望用户侧增强向量 $\mathbf{a}_u$ 能够学习用户历史上可能匹配的物品 Embedding，而物品侧增强向量 $\mathbf{a}_v$ 能够学习物品历史上可能匹配的用户 Embedding。这样，增强向量就可以被理解为对另一侧历史交互信息的高层语义摘要。

对于一个用户、物品训练样本 (u, v, y)，当用户与物品之间存在正向反馈，即 $y = 1$ 时，希望用户增强向量 $\mathbf{a}_u$ 能够逼近该物品对应的 Embedding $\mathbf{p}_v$；反过来，希望物品增强向量 $\mathbf{a}_v$ 能够逼近该用户对应的 Embedding $\mathbf{p}_u$。因此，可以定义用户侧和物品侧的模仿损失（Mimic Loss）为：

$$\mathcal{L}_u = \frac{1}{T} \sum_{(u,v,y=1)\in\mathcal{T}} y\|\mathbf{a}_u - \mathrm{sg}(\mathbf{p}_v)\|_2^2, \quad \mathcal{L}_v = \frac{1}{T} \sum_{(u,v,y=1)\in\mathcal{T}} y\|\mathbf{a}_v - \mathrm{sg}(\mathbf{p}_u)\|_2^2 \tag{11.18}$$

其中 $\mathrm{sg}(\cdot)$ 表示 Stop Gradient 操作。

需要注意的是，这里的 Stop Gradient 机制非常重要。由于模仿损失的目标是训练增强向量 $\mathbf{a}_u$ 和 $\mathbf{a}_v$ 去拟合另一侧已经学习到的 Embedding，因此不希望模仿损失反向修改作为监督目标的 $\mathbf{p}_u$ 和 $\mathbf{p}_v$。通过 Stop Gradient，可以使梯度仅更新增强向量相关参数，而不会通过模仿损失反向传播到另一侧的 Embedding 网络中。

从直观上看，对于一个产生正向反馈的用户、物品 Pair 对，用户增强向量会不断吸收用户历史交互物品的 Embedding 信息，而物品增强向量则不断吸收与其产生交互的用户 Embedding 信息。因此，$\mathbf{a}_u$ 可以理解为用户兴趣在物品表示空间中的一种压缩表达，而 $\mathbf{a}_v$ 则可以理解为物品在用户表示空间中的一种历史交互表达。

通过引入上述自适应模型机制和对应的损失函数，DAT 模型可以在不破坏双塔模型基本结构的前提下，实现一种隐式的信息交互，即：一侧塔通过增强向量获得另一侧历史交互信息，再利用这些信息辅助当前塔学习更加具有判别性的 Embedding 表征。这种方法相比起直接增加用户与物品的交叉特征，更加适合需要大规模向量检索以及在线高效推理的工业推荐场景。

11.1.4.3 类别对齐损失

除了利用增强向量来改善双塔模型的特征交互能力之外，DAT 模型还针对工业级推荐场景中的类别数据不均衡问题提出了类别对齐的损失（Category Alignment Loss，CAL）。从具体业务视角来看，在实际推荐系统中，

不同类别的物品数量通常存在明显的差异。比如，某些热门类别可能拥有大量训练样本，而一些长尾类别的数据规模则非常有限。在这种数据分布不均衡的情况下，双塔模型往往会优先学习数据量较大类别的表示模式，而数据量较小的类别由于缺乏充分的训练样本，其 Embedding 表示可能更加不稳定，从而导致模型在不同类别上的性能存在较大差异。

DAT 模型利用数据量最大的类别作为主要类别（Major Category），并将其他类别的 Embedding 表示与主要类别的表示分布进行对齐。具体来说，假设一个 Batch 中数据量最大的类别对应的物品 Embedding 集合为 $\mathcal{S}_{major}$，其他类别对应的 Embedding 集合分别为 $\mathcal{S}_2, \mathcal{S}_3, \ldots, \mathcal{S}_n$，则 DAT 模型通过比较不同类别 Embedding 的协方差矩阵来衡量其表示分布之间的差异。对于 Embedding 集合 $\mathcal{S}$，其协方差矩阵可以表示为

$$\mathbf{C}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{p} \in \mathcal{S}} (\mathbf{p} - \boldsymbol{\mu})(\mathbf{p} - \boldsymbol{\mu})^\top \tag{11.19}$$

其中 $\boldsymbol{\mu}$ 表示 Embedding 集合的均值。进一步地，类别对齐损失（Category Alignment Loss，CAL）可以定义为：

$$\mathcal{L}_{CA} = \sum_{i=2}^{n} |\mathbf{C}(\mathcal{S}_{major}) - \mathbf{C}(\mathcal{S}_i)|_F^2 \tag{11.20}$$

通过最小化上述损失，不同类别的 Embedding 表示在二阶统计特征上能够更加接近，从而将数据量较大类别中学习到的表示结构迁移到数据量较小的类别。需要注意的是，CAL 并不是简单地让不同类别的 Embedding 向量逐点相同，而是对不同类别 Embedding 的整体分布进行约束，因此能够在一定程度上保留不同类别自身的语义差异。

11.1.4.4 DAT 模型的训练目标

DAT 模型将推荐检索问题建模为二分类问题，并采用随机负采样构造训练样本。对于每一个正向的用户、物品交互 (u, v^+)，从全量物品集合中随机采样 S 个未交互物品作为负样本，从而形成一个包含 $S+1$ 个候选物品的训练集合。DAT 模型首先利用用户 Embedding 和物品 Embedding 之间的内积计算匹配分数，再通过 Sigmoid 函数将其转换为正向反馈概率。对于训练样本 (u, v, y)，其预测损失可以表示为：

$$\mathcal{L}_p = -\frac{1}{T} \sum_{(u,v,y) \in \mathcal{T}} [y \log \sigma(s(u,v)) + (1-y) \log(1 - \sigma(s(u,v)))] \tag{11.21}$$

其中 T 表示训练样本总数。最终，DAT 模型将匹配任务、自适应模仿机制以及类别对齐损失函数联合起来进行训练。因此，DAT 模型整体的优化目标为：

$$\mathcal{L} = \mathcal{L}_p + \lambda_1 \mathcal{L}_u + \lambda_2 \mathcal{L}_v + \lambda_3 \mathcal{L}_{CA} \tag{11.22}$$

其中 λ_1、λ_2 和 λ_3 分别控制用户侧模仿损失、物品侧模仿损失以及类别对齐损失的权重。

从整个训练过程来看，DAT 模型实际上通过三个层面的优化增强了传统双塔模型。第一个层面是基础的匹配目标 $\mathcal{L}_p$，直接优化用户与物品之间的匹配准确性；第二层是 AMM，通过 $\mathcal{L}_u$ 和 $\mathcal{L}_v$ 将用户与物品的历史交互信息注入对应的增强向量，使两个塔能够间接获得另一侧的信息；第三层是 CAL，通过对不同类别 Embedding 分布进行对齐，缓解类别数据不均衡带来的表示学习偏差。

11.1.4.5 DAT 模型的工业价值

DAT 模型的核心价值在于，它并没有通过直接引入用户与物品的交叉网络来换取表达能力，而是在保留双塔模型在线向量匹配范式的基础上，通过引入增强向量来实现两个塔之间的隐式信息交互。因此，DAT 模型在模型表达能力和在线检索效率之间取得了较好的平衡。

与传统双塔模型相比，DAT 模型通过 AMM 引入了另一侧的历史交互信息，使用户 Embedding 不再完全局限于用户自身特征，物品 Embedding 也不再完全局限于物品自身特征，从而能够学习更加丰富的用户与物品之间的关系。与此同时，最终的匹配函数仍然是用户 Embedding 与物品 Embedding 之间的内积，因此物品 Embedding 仍然可以离线或准实时预计算，并利用 ANN 索引进行大规模候选检索。虽然 DAT 模型在论文中主要是应用在召回场景中进行双塔模型的改进，但其改进的思想仍然是可以在粗排的双塔模型中进行落地应用。因此在本书

内容编排的时候，将 DAT 模型放在粗排章节中进行介绍，用于提供一种新的强化粗排双塔模型信息交互能力的解决方案。

从工业级推荐系统的角度来看，DAT 模型可以理解为在不改变双塔模型核心推理范式的情况下，对双塔模型进行的一种**“表征增强”**。这也是该方法与显式交叉模型的一个重要区别：显式交叉模型通常需要在在线阶段针对每一个用户、物品 Pair 对进行特征交互计算，而 DAT 模型主要通过训练阶段学习增强向量，使最终在线推理仍然能够保持向量匹配的高效形式。因此，DAT 模型尤其适合对在线延迟和候选规模具有严格要求的大规模工业推荐系统。

11.2 多塔模型

在前面的双塔模型章节中，我们已经介绍了 DSSM、TDSSM、DAT 等经典双塔模型。传统双塔模型通过分别学习用户表示和物品表示，再利用向量匹配完成候选排序，具有结构简单、计算效率高、易于在线部署等优点，因此长期以来一直是工业级粗排模型的主流架构。然而，随着推荐系统业务规模的不断扩大以及建模目标日益复杂，传统双塔模型逐渐暴露出表达能力有限的问题。一方面，推荐系统需要同时服务于短视频、直播、电商、本地生活等多种业务，不同业务往往具有不同的物品特征分布和兴趣模式；另一方面，仅依赖用户塔与物品塔的独立编码，难以充分建模用户与物品之间复杂的高阶交互关系。此外，在工业实践中还需要考虑购买、支付等延迟反馈行为，而传统双塔模型通常采用统一的物品表示，难以同时兼顾即时反馈与延迟反馈任务。

因此，近年来工业界逐渐从传统双塔模型演进到多塔模型（Multi-Tower Model）。这里所说的“塔（Tower）”，本质上是针对某一类特征、某一种业务目标或者某一类物品构建的独立表示学习网络。不同 Tower 分别负责学习不同的信息表示，最后再通过向量融合、特征拼接或加权组合等方式共同完成最终预测。相比传统双塔模型，多塔模型能够进一步增强模型表达能力，同时兼顾不同业务场景下的建模需求，因此已经成为当前工业级粗排模型的重要发展方向。

总体而言，工业界多塔模型的发展主要经历了以下几种典型演进方式：

- **增加物品塔（Multiple Item Towers）**：针对不同类型的物品建立独立的物品塔，共享用户兴趣表示，实现跨业务、多领域的统一建模，代表模型包括 MV-DSSM 模型。
- **增加延迟反馈塔（Delayed Feedback Tower）**：针对购买、支付等延迟反馈行为，单独建立对应的物品塔。其主要目的并非为了支持多任务学习，而是由于即时反馈与延迟反馈在数据流组织、样本构建以及标签成熟时间等方面存在明显差异，统一使用同一个物品表示往往难以兼顾两类反馈。因此，工业界通常会为延迟反馈建立独立的物品表示学习网络，从而更好地建模长期转化行为。
- **增加辅助特征塔（Feature Towers）**：在传统双塔模型之外增加用于学习辅助特征或显式交叉特征的表示学习网络，例如 FM 塔、交叉特征塔（Cross Interaction Tower）等，以增强模型对于用户与物品高阶交互关系的建模能力。

下面将分别介绍工业界最具代表性的几类多塔模型及其工程实现方式。

11.2.1 MV-DSSM 模型

MV-DSSM（Multi-View Deep Structured Semantic Model）模型是微软于 2015 年提出的一种经典双塔模型，源自于 WWW 会议论文《A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems》[2]。相比其原始的 DSSM 模型只能建模单一类型用户-物品匹配关系，MV-DSSM 模型进一步提出了**多视图（Multi-View）语义学习框架**，能够同时学习用户与多种类型物品之间的匹配关系，因此成为后来跨域推荐（Cross-domain Recommendation）以及多场景推荐（Multi-Scenario Recommendation）的重要基础模型。

原始 DSSM 模型采用双塔结构，仅包含一个用户塔（User Tower）和一个物品塔（Item Tower），主要用于学习一种类型用户与一种类型物品之间的语义匹配关系。然而，在工业推荐系统中，平台通常同时存在多种不同类型的推荐内容。例如，短视频平台既需要推荐短视频，也需要推荐直播、图文以及商品；综合内容平台还可

能同时包含新闻、音乐、游戏等多个业务域。这些不同类型的物品往往具有不同的特征体系，其输入特征维度、Embedding 表示以及语义空间均存在较大差异，因此难以直接利用同一个 DSSM 模型进行统一建模。

一种最直接的解决方案是针对每一种物品类型分别训练一个独立的 DSSM 模型。例如，对于短视频、直播以及图文内容，可以分别训练三个双塔模型，每个模型独立学习对应业务域的用户兴趣表示。然而，这种方案存在以下几个问题：

- 不同业务域分别学习独立的用户塔，大量重复建模用户兴趣，不仅增加模型参数规模，也带来了额外的训练和部署成本。
- 用户在不同业务域中的行为本质上反映的是同一用户的兴趣偏好，各业务域之间存在大量潜在关联，而多个独立模型无法充分利用这种跨域监督信息。
- 对于数据规模较小的新业务，由于用户行为样本不足，单独训练双塔模型容易出现过拟合，模型泛化能力较差。

针对上述问题，MV-DSSM 模型提出了共享用户表示（Shared User Representation）的思想，即多个业务域共享同一个用户塔，而针对每一种物品类型分别构建独立的物品塔，从而形成“单用户塔、多物品塔”的网络结构，如图 11.5所示。

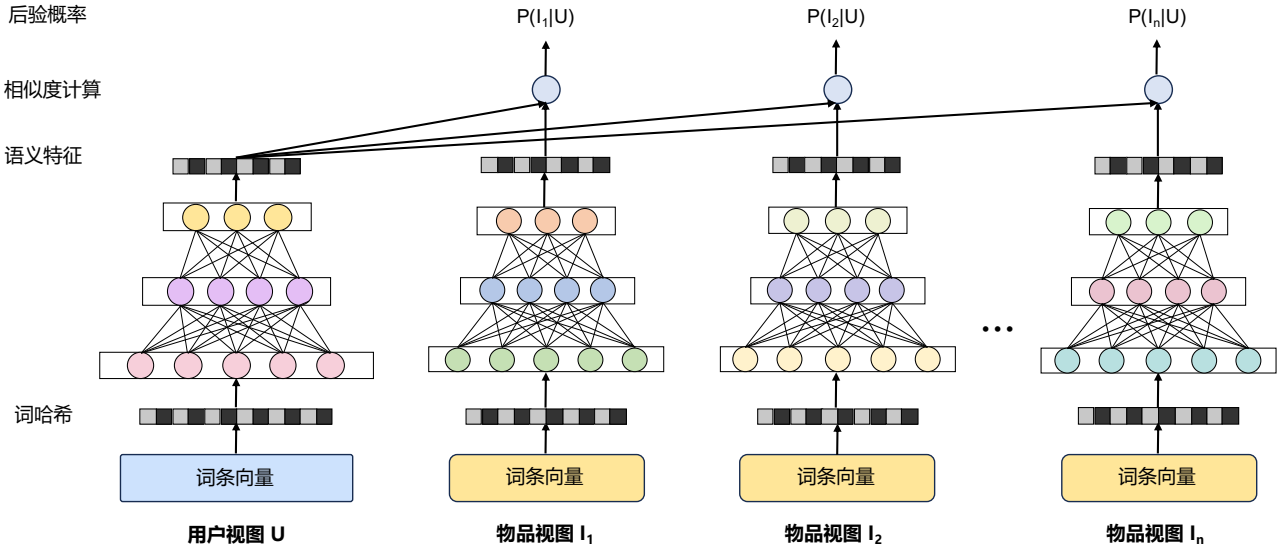

图 11.5: MV-DSSM 模型整体结构示意图。

假设共享用户塔表示为

$$\mathbf{e}_u = f_u(\mathbf{x}_u) \tag{11.23}$$

其中 $\mathbf{x}_u$ 表示用户输入特征，$f_u(\cdot)$ 表示共享用户编码网络。对于第 m 个业务域，其对应的物品塔可表示为

$$\mathbf{e}_i^{(m)} = f_i^{(m)}(\mathbf{x}_i^{(m)}) \tag{11.24}$$

其中 $f_i^{(m)}(\cdot)$ 表示第 m 个业务域对应的物品编码网络，不同业务域拥有各自独立的网络参数。随后，每个业务域分别计算用户 Embedding 与对应物品 Embedding 之间的余弦相似度：

$$s_m(u,i) = \cos\left(\mathbf{e}_u, \mathbf{e}_i^{(m)}\right) \tag{11.25}$$

在训练过程中，每一个训练样本仅激活对应业务域的物品塔。例如，当训练样本来自直播推荐任务时，仅更新共享用户塔以及直播物品塔，而短视频、图文等其他物品塔保持不变。由于所有业务域均共享同一个用户塔，因此来自不同业务域的监督信号都会共同优化用户表示，使得最终学习得到的用户 Embedding 能够同时适用于多个推荐场景。

从优化目标来看，MV-DSSM 模型本质上仍然继承了 DSSM 模型的点击监督训练方式。对于每一个业务域，模型最大化用户与点击物品之间的相似度，同时最小化用户与负样本之间的相似度，不同业务域交替进行随机

梯度下降（SGD）训练，从而共同学习共享的用户语义空间。相比起原始的 DSSM 模型，MV-DSSM 模型通常具有以下几个方面的优势：

- **共享用户兴趣建模。**多个业务域共享同一个用户塔，避免了重复学习用户表示，显著减少模型参数规模，同时提升了用户 Embedding 的泛化能力。
- **充分利用跨域监督信息。**不同业务域产生的点击行为共同参与用户表示学习，使得数据量较大的业务域能够帮助数据较少的业务域学习更加鲁棒的用户兴趣表示，实现知识迁移（Knowledge Transfer）。
- **支持异构输入特征。**与原始 DSSM 模型主要面向文本检索不同，MV-DSSM 模型取消了 Letter Trigram 等输入限制，每个业务域可以根据自身特点采用不同类型的输入特征，包括离散类别特征、连续数值特征、地理位置特征以及文本特征等，更符合工业推荐系统异构特征建模需求。
- **易于扩展新的业务域。**当系统新增一种推荐内容时，仅需增加对应的物品塔即可，无需重新训练已有业务模型，因此具有较好的可扩展性。

MV-DSSM 模型提出的共享用户塔、多物品塔架构，对后续工业推荐系统的发展产生了重要影响。目前，多业务推荐、多场景推荐以及跨域推荐中的大量双塔模型均采用了类似的设计思想。例如，在短视频推荐系统中，可以利用共享用户表示同时建模短视频、直播以及商城等多个业务；在综合内容平台中，也可以通过共享用户塔统一学习新闻、视频、音乐等不同内容形态的用户兴趣，从而进一步提升整体推荐效果。因此，MV-DSSM 模型不仅是 DSSM 模型的重要扩展，也是现代多场景推荐模型的重要基础。

除此之外，MV-DSSM 模型实际上也启发了工业级推荐系统粗排模型中的多塔模型（Multi-Tower Model）设计思想。多塔模型通常在粗排阶段同时建模用户兴趣、物品特征以及上下文信息等多个维度，通过不同的塔结构分别学习各类特征表示，并在最后进行特征融合与匹配计算，从而提升粗排候选的整体质量。相比单一双塔模型，多塔模型能够更好地捕捉复杂的用户兴趣模式以及多样化的物品特征，进一步提高粗排阶段的召回精度和覆盖率。

11.2.2 DSSM + FM 模型

在工业级推荐系统的粗排阶段，模型不仅需要具备较高的预测精度，还需要满足严格的在线时延约束。因此，工业界通常采用以 DSSM 为主体、FM（Factorization Machine）为辅助的混合建模方式，即 DSSM+FM 模型。该方法利用 DSSM 学习用户与物品的高层语义表示，同时引入 FM 模块显式建模用户与物品之间的重要二阶交叉特征，在几乎不显著增加在线计算开销的前提下，进一步提升粗排模型的整体效果，因此成为工业界粗排模型中一种十分常见的网络结构。

传统 DSSM 本质上属于双塔模型，用户侧和物品侧分别经过独立的神经网络进行编码，两侧直到最后才通过向量内积或余弦相似度进行匹配。这种结构具有计算效率高、易于部署等优点，但由于用户塔和物品塔之间缺乏显式特征交互，因此模型难以学习一些具有明确业务意义的重要组合特征。例如，不同用户对于不同作者、不同商品、不同设备环境下的点击偏好往往存在明显差异，而仅依赖 DSSM 学习得到的语义 Embedding，很难充分刻画这些细粒度交互关系。

因此，工业级推荐系统的粗排模型迭代中一个很关键的研究问题就是：

命题 11.1 (DSSM 模型的显式交互建模问题)

DSSM 模型由于用户塔与物品塔之间缺乏显式特征交互，难以学习一些具有明确业务意义的重要组合特征，从而影响粗排模型的整体效果。那么如何在保持 DSSM 高效计算能力的前提下，进一步提升模型对重要离散特征的显式交互建模能力，从而提升粗排模型的整体性能呢？

针对这一问题，工业界通常会在 DSSM 主网络之外增加一个 FM 辅助网络的分支，对部分重要离散特征进行二阶交叉建模，如图 11.6所示。

整个模型主要由两个部分组成：

- **DSSM 主网络。**用户特征与物品特征分别经过用户塔（User Tower）和物品塔（Item Tower）编码，学习用户 Embedding 与物品 Embedding，并利用向量相似度刻画用户与物品之间的整体语义匹配关系。

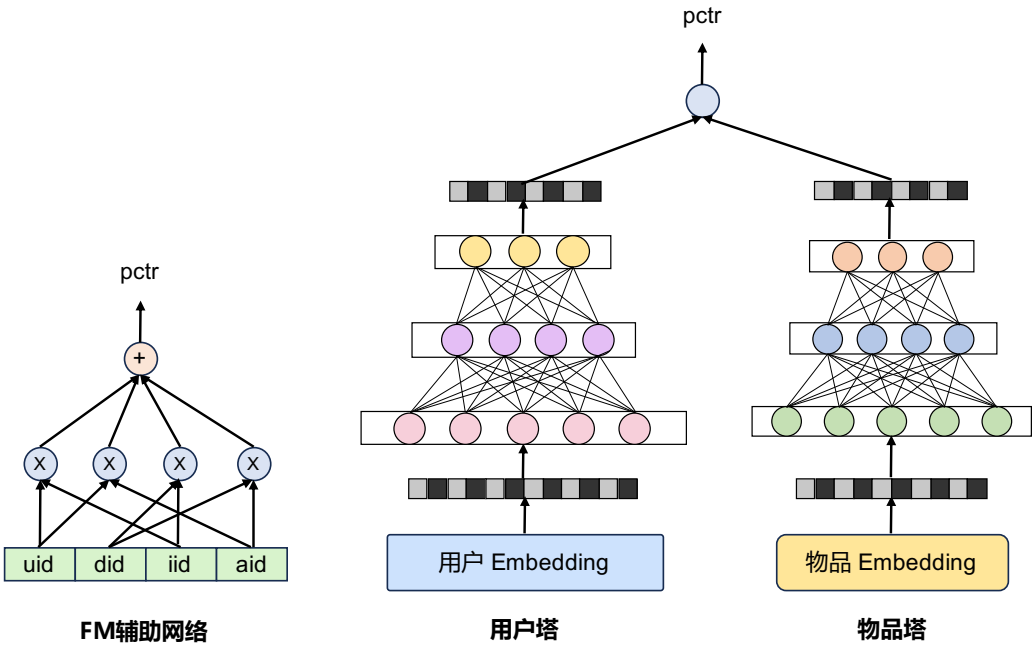

图 11.6: DSSM+FM 模型整体结构示意图。

- **FM 交叉网络**。针对少量具有较强业务意义的重要离散特征，利用 FM 显式学习二阶特征交互，从而弥补双塔模型缺乏显式交互建模能力的不足。

假设用户 Embedding 与物品 Embedding 分别表示为

$$\mathbf{e}_u = f_u(x_u), \qquad \mathbf{e}_i = f_i(x_i), \tag{11.26}$$

则 DSSM 部分通常采用内积或余弦相似度计算用户与物品之间的语义匹配分数：

$$s_{\text{DSSM}} = \mathbf{e}_u^\top \mathbf{e}_i \tag{11.27}$$

或

$$s_{\text{DSSM}} = \cos(\mathbf{e}_u, \mathbf{e}_i) \tag{11.28}$$

另一方面，FM 模块主要针对离散 ID 特征进行二阶交叉建模。设输入特征向量表示为

$$\mathbf{x} = (x_1, x_2, \cdots, x_n) \tag{11.29}$$

则 FM 模型可表示为

$$s_{\text{FM}} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \tag{11.30}$$

其中，$\mathbf{v}_i$ 表示第 i 个离散特征对应的隐向量（Latent Vector），$\langle \cdot, \cdot \rangle$ 表示向量内积。相比传统人工构造交叉特征，FM 能够自动学习不同离散特征之间的重要组合关系，因此具有较好的泛化能力。

在工业实践中，FM 模块通常不会覆盖所有输入特征，而是仅选择少量最重要的 ID 特征参与交叉建模。例如，较为常见的一组输入包括：用户 ID（User ID）、用户设备 ID（Device ID）、物品 ID（Item ID）、作者 ID（Author ID）。这些特征之间通常存在较强的业务相关性。例如，不同用户对于不同作者具有明显不同的兴趣偏好，不同设备用户对于内容消费行为也可能存在差异，因此利用 FM 学习这些 ID 之间的二阶交叉特征，往往能够获得较为稳定的效果提升。

在获得 DSSM 模型分支和 FM 辅助网络分支的输出之后，两部分结果会分别预估具体业务场景中关键的用户反馈信号，例如点击、收藏、购买等。模型训练阶段仍然采用点击日志作为监督信号，以点击、播放、收藏、点赞、购买等用户行为作为正样本，通过交叉熵损失或 Pairwise Loss 对整个网络进行端到端优化，使 DSSM 主网络与 FM 交叉网络共同学习用户与物品之间的匹配关系。

为了线上推理的计算效率，如果仍然采用双塔模型的方式，通常 DSSM 主网络中的物品塔会通过预处理的方式提前计算，而在用户请求到来时，仅在线计算用户塔的 Embedding，并与预先计算好的物品 Embedding 进行点积运算来得到最终的 DSSM 网络预估分。为了不增加线上推理的耗时，FM 模块通常仅在模型训练时作为辅助网络参与优化，而在推理阶段不参与计算。也就是说，FM 模块主要用于提升 DSSM 主网络的 Embedding 学习能力，从而间接提升 DSSM 主网络的整体性能。但如果线上的耗时压力不大，由于 FM 模块的计算量相对较小，也可以在推理阶段同时计算 DSSM 主网络和 FM 模块的输出，并将两者的结果进行融合，从而进一步提升粗排模型的整体性能。

值得注意的是，在工业实践中，FM 模块中的 ID 特征 Embedding 通常会与 DSSM 主网络对应的 Embedding 进行参数共享。例如，用户 ID、物品 ID、作者 ID 等 Embedding 既作为 DSSM 主网络的输入，也作为 FM 模块中对应特征的隐向量。由于两部分网络共同优化同一组 Embedding，因此 FM 学习到的细粒度交叉信息能够反向促进 DSSM 主网络的 Embedding 学习，而 DSSM 学习得到的全局语义表示也能够提升 FM 特征的泛化能力。这种 Embedding 共享策略在许多工业系统中能够取得较好的效果，因此成为一种较为常见的工程实践。

然而，Embedding 共享并非始终能够带来收益。当数据噪声较大、训练样本较少或不同网络分支的优化目标存在较大差异时，共享 Embedding 也可能导致梯度相互干扰（Gradient Interference），使不同任务之间产生负迁移（Negative Transfer），反而降低模型整体性能。例如，当 FM 分支更加关注局部 ID 交叉关系，而 DSSM 主网络更强调全局语义表示时，两者学习方向可能并不完全一致，共享 Embedding 可能导致两部分网络相互制约。因此，在实际工业推荐系统中，是否采用 Embedding 共享通常需要结合具体业务场景，通过离线实验以及线上 A/B 测试进行验证，而不能作为固定的模型设计原则。归根结底，工业推荐系统模型的设计是一门实验科学，许多网络结构与优化策略都需要结合具体的数据分布、业务特点以及线上反馈持续迭代优化，而不存在适用于所有场景的统一最优方案。

11.2.3 延迟反馈多塔模型

除了针对不同类型物品建立多个物品塔之外，近年来工业界另一类典型的多塔建模方式，是针对延迟反馈（Delayed Feedback）建立独立的物品表示学习网络。相比 MV-DSSM 主要解决不同物品类型之间的统一表示学习问题，延迟反馈多塔模型更加关注推荐系统中由于标签生成时序不同所带来的样本组织、模型训练以及表示学习问题。

在工业级推荐系统中，不同用户行为具有完全不同的反馈时效。例如点击、播放、点赞、收藏等行为通常发生在曝光后的数秒钟以内，因此属于即时反馈（Immediate Feedback）；而购买、支付、续费、长期留存等行为往往发生在曝光后的数小时、数天甚至更长时间之后，因此属于延迟反馈（Delayed Feedback）。

延迟反馈最大的特点在于标签成熟时间（Label Maturity）明显晚于样本产生时间。当一条曝光样本进入训练系统时，即时反馈标签通常已经可以获得，而延迟反馈标签此时往往仍然未知。如果直接将尚未发生购买、支付等行为的样本视为负样本参与训练，则未来真正发生转化时，这些样本实际上属于错误监督（False Negative）；而如果始终等待延迟标签成熟之后再统一生成训练样本，又会导致模型无法及时学习最新曝光数据，降低模型更新效率。因此，如何组织训练样本、如何设计模型训练流程成为延迟反馈建模中的核心问题。

另一方面，即时反馈与延迟反馈本身所关注的用户兴趣也存在明显差异。点击、播放等即时行为更多反映用户当前时刻的短期兴趣，而购买、支付等延迟行为则更加依赖商品质量、价格区间、商家信誉以及用户长期消费能力等因素。因此，两类监督信号对应的物品表示往往具有不同的优化方向。如果完全共享同一套物品表示，容易导致不同目标之间产生梯度冲突（Gradient Conflict），影响模型整体效果。

基于上述原因，工业界通常会在传统双塔模型基础上增加一条独立的延迟反馈物品塔（Delayed Item Tower），如图 11.7所示。整个模型仍然共享同一套用户塔，而物品侧则分别学习即时反馈物品表示与延迟反馈物品表示，从而兼顾短期行为预测与长期价值建模。即时反馈物品塔主要学习点击、播放等行为对应的物品表示：

$$\mathbf{e}_{item}^{click} = f_{click}(\mathbf{x}_{item}) \tag{11.31}$$

其中 $\mathbf{x}_{item}$ 通常包括物品 ID、作者 ID、类目、标签、文本 Embedding、图像 Embedding、视频 Embedding 等基础物品特征。延迟反馈物品塔则更加关注能够影响长期转化行为的信息，例如商品属性、价格区间、商家质量、

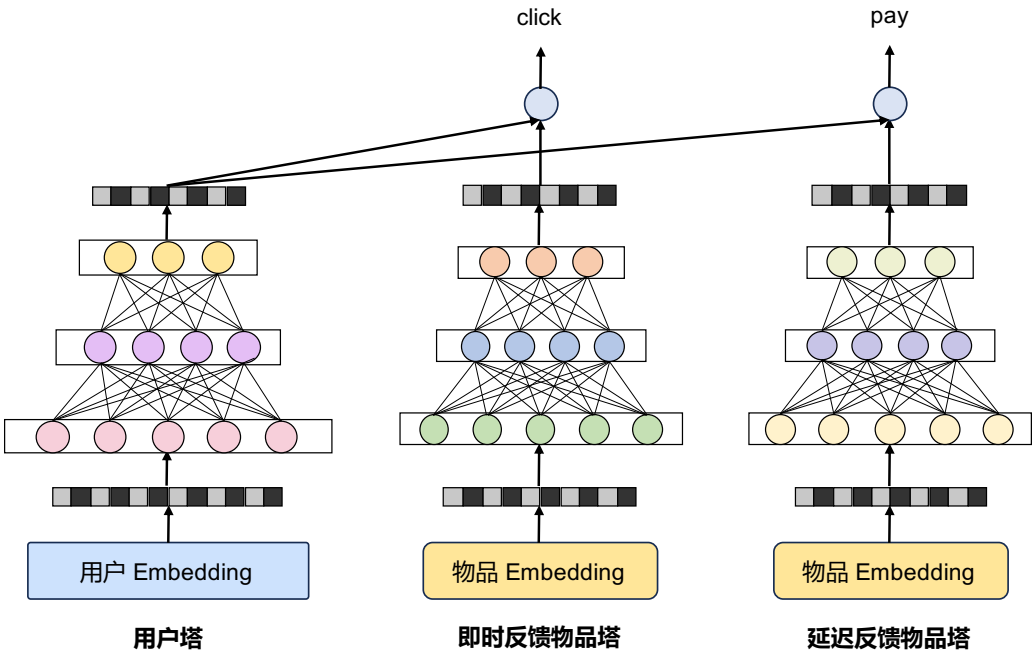

图 11.7: 延迟反馈多塔模型整体结构示意图。

历史成交率、长期内容质量以及其他稳定特征，其表示可写为：

$$\mathbf{e}_{item}^{delay} = f_{delay}(\mathbf{x}_{item}) \tag{11.32}$$

随后，即时反馈的物品塔和延迟反馈的物品塔输出的两个物品 Embedding 分别与共享用户塔输出的用户 Embedding 通过点积运算完成匹配：

$$s_{click} = \mathbf{e}_{user}^{\top}\mathbf{e}_{item}^{click}, \qquad s_{delay} = \mathbf{e}_{user}^{\top}\mathbf{e}_{item}^{delay} \tag{11.33}$$

这里需要注意的是，如果延迟反馈信号与即时反馈信号之间存在较强的相关性，则可以在两个物品塔之间共享部分底层网络参数，从而进一步提升模型的泛化能力。然而如果两类反馈信号之间差异较大，则通常会为延迟反馈物品塔单独设计一套网络结构，以避免不同目标之间的梯度冲突。

这类带有延迟反馈的多塔模型，其训练过程相比普通双塔模型更加复杂。根据训练数据流的组织方式不同，目前工业界主要存在两种典型实现方式：一种是基于 Kafka、Flink 等实时数据流的在线增量训练；另一种是基于 Hive、HDFS 等离线数据仓库的周期训练。下面分别介绍两种训练方式。

11.2.3.1 实时数据流下的延迟反馈建模

对于采用 Kafka、Flink 等实时数据流训练的工业级推荐系统，模型通常采用在线增量训练（Online Incremental Training）的方式持续更新参数。相比离线训练，实时训练最大的特点在于能够快速学习最新曝光数据，及时反映用户兴趣变化，因此目前已经广泛应用于短视频推荐、直播推荐、信息流推荐以及广告推荐等对模型实时性要求较高的场景。

然而，实时训练同样带来了延迟反馈建模的困难。当一条曝光日志进入实时训练系统时，点击、播放等即时反馈通常已经产生，因此可以立即构造监督信号；而购买、支付、续费等延迟反馈往往需要等待数分钟、数小时甚至数天之后才能最终确定。如果始终等待延迟反馈标签成熟之后再生成训练样本，则模型参数更新将严重滞后，难以及时学习最新曝光数据。因此，工业界通常不会等待延迟标签成熟，而是采用事件驱动（Event-driven）的训练方式，将曝光事件和后续行为事件分别作为两次模型更新。

具体而言，当曝光之后用户产生点击、播放等即时反馈时，系统首先生成一条实时训练样本。由于延迟反馈

尚未发生，因此对应延迟标签通常初始化为负样本，即

$$y_{delay} = 0 \tag{11.34}$$

此时，该样本进入统一的实时训练任务（Training Job），同时参与即时反馈 Tower 与延迟反馈 Tower 的训练。其中，即时反馈 Tower 使用真实点击标签进行监督，而延迟反馈 Tower 则暂时将该样本视为未发生延迟反馈的样本进行优化。

随后，如果未来该曝光最终产生购买、支付等延迟行为，则行为日志再次进入实时训练数据流。系统根据曝光 ID、用户 ID 以及物品 ID 重新定位对应曝光样本，并再次构造训练样本，此时延迟标签更新为

$$y_{delay} = 1 \tag{11.35}$$

模型再次利用该样本完成一次参数更新，从而实现标签修正（Label Correction）。因此，同一条曝光样本在实时训练过程中通常会经历两次训练：第一次利用即时反馈进行训练；第二次在延迟标签成熟之后再次进行训练，以学习长期目标。

需要注意的是，整个训练过程通常仍然对应一个统一的训练任务，而不是分别维护两个独立模型。实时曝光样本以及后续到达的延迟反馈样本都会进入同一条训练数据流，共享同一个用户塔，如图 11.7所示。对于不同塔而言，其参数更新方式有所区别：

- 用户塔始终参与所有训练样本更新，从而持续学习最新用户兴趣表示；
- 点击物品塔仅利用即时反馈监督信号进行更新；
- 延迟反馈物品塔仅利用延迟反馈标签对应的损失进行更新。

假设即时反馈损失与延迟反馈损失分别表示为

$$L_{click} = \ell(\hat{y}_{click}, y_{click}), \qquad L_{delay} = \ell(\hat{y}_{delay}, y_{delay}) \tag{11.36}$$

其中，ℓ 表示损失函数，$\hat{y}_{click}$ 表示即时反馈预测值，y_{click} 表示即时反馈真实标签；$\hat{y}_{delay}$ 表示延迟反馈预测值，y_{delay} 表示延迟反馈真实标签。则整个模型通常采用联合优化目标：

$$L = L_{click} + \lambda L_{delay} \tag{11.37}$$

其中 λ 表示延迟反馈任务对应的损失权重。由于用户塔同时参与两个任务，因此其参数能够同时学习短期兴趣与长期兴趣；而两个物品塔则分别学习各自业务目标对应的物品表示，从而避免不同监督目标直接作用于同一套物品 Embedding 所产生的梯度冲突。

不过，上述训练方式仍然存在一个经典问题，即 **False Negative（伪负样本）问题**。对于第一次进入训练的数据而言，虽然当前尚未发生购买行为，但其中一部分样本未来仍然可能完成转化，因此这些样本实际上属于“暂时未发生购买”，而并非真正不会购买。如果直接采用普通负样本进行训练，则会导致模型低估未来可能发生转化的样本，产生监督偏差。

针对这一问题，近年来工业界提出了大量延迟反馈学习（Delayed Feedback Learning）方法，其中最具代表性的包括 Delayed Feedback Model（DFM）[1]、Entire Space Delayed Feedback Model（ESDF）[10] 以及 Fake Negative Weighting（FNW）[5] 等方法。

11.2.3.1.1 （1）Delayed Feedback Model（DFM）

DFM 是最早针对延迟反馈问题提出的建模方法之一，其核心思想是在建模最终转化概率的同时，引入用户转化时间分布（Conversion Delay Distribution）的估计。模型不仅预测用户是否最终发生购买，还同时预测购买将在什么时候发生，从而对尚未成熟的样本进行更加合理的概率估计，而不是简单地将其视为负样本。然而，由于需要额外建模时间分布，DFM 的训练过程相对复杂，并且需要较长时间窗口统计延迟分布，因此工程实现成本较高。

11.2.3.1.2 （2）Entire Space Delayed Feedback Model（ESDF）

ESDF 模型进一步提出，不应仅利用已经成熟标签进行训练，而应利用整个曝光空间（Entire Space）上的样本共同学习点击概率与转化概率。假设曝光事件

为 E，点击事件为 C，最终转化事件为 V，则最终转化概率可以写为

$$P(V|E) = P(C|E) \times P(V|C,E) \tag{11.38}$$

其中第一部分对应 CTR 模型，第二部分对应 CVR 模型。ESDF 模型利用所有曝光样本共同优化整个概率空间，而不是仅利用点击样本训练 CVR 模型，因此能够有效缓解由于样本删失（Sample Selection Bias）以及延迟反馈带来的偏差，目前已经成为广告推荐领域最经典的延迟反馈学习方法之一。

11.2.3.1.3 （3）Fake Negative Weighting（FNW） 相比起重新设计整个概率模型，工业界更加广泛采用的方法是直接降低伪负样本对于模型训练的影响。FNW 认为，曝光时间较短的负样本未来仍然具有较大的转化概率，因此不应给予普通负样本相同的监督强度。通常采用时间相关权重函数重新加权 Loss：

$$L_{delay} = w_t \ell(\hat{y}_{delay}, y_{delay}) \tag{11.39}$$

其中 t 表示曝光距离当前的时间间隔。当曝光时间较短时，w_t 通常较小，以减弱可能发生转化样本带来的错误监督；随着曝光时间不断增加，负样本最终发生转化的概率逐渐降低，此时 w_t 逐渐增大，并最终接近普通负样本对应的监督权重。由于 FNW 无需改变整体模型结构，仅通过 Loss 重加权即可完成训练，因此实现简单、计算代价较低，目前已经成为工业界实时训练系统中最常见的延迟反馈优化策略之一。

总体来看，实时数据流下的延迟反馈建模，本质上是在模型实时更新能力与监督标签准确性之间进行权衡。工业界通常采用统一训练任务结合多塔结构，通过事件驱动的数据流持续更新模型参数，再结合 ESDF、FNW 等延迟反馈学习方法缓解 False Negative 问题，从而在保证模型实时性的同时兼顾长期业务目标。

11.2.3.2 离线数据流下的延迟反馈建模

相比起实时增量的训练方式，另一类更加广泛应用于工业推荐系统的是基于 Hive、HDFS 等离线数据仓库的周期训练（Offline Periodic Training）。该方案通常按照小时级（Hourly）或天级（Daily）的周期生成训练样本，再利用分布式训练框架完成模型参数更新。相比起实时训练更加关注模型的新鲜度（Freshness），离线训练则更加关注训练标签的完整性（Label Completeness）以及模型收敛的稳定性，因此在电商推荐、广告推荐、本地生活推荐等长期价值优化场景中得到广泛应用。

离线训练最大的优势在于能够充分等待延迟反馈标签成熟之后再生成训练样本。例如，对于曝光发生在第 T 天的样本，在生成第 $T+1$ 天训练数据时，大部分点击、播放、点赞等即时反馈已经全部产生，同时购买、支付等短周期延迟反馈通常也已经完成，因此可以直接获得更加准确的监督标签，而无需像实时训练那样首先将样本视为负样本，再等待后续标签修正。

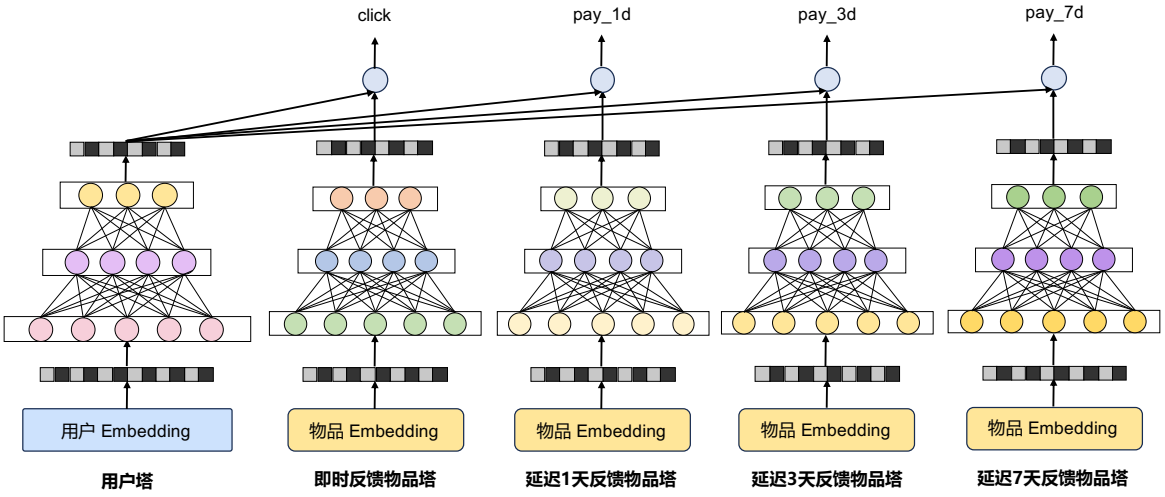

图 11.8: 离线训练下的延迟反馈多塔模型整体结构示意图。

与此同时，离线训练能够方便地访问历史多个时间窗口的数据，因此工业界通常会构建多个标签成熟窗口（Label Window）对应的训练数据集。例如，以曝光时间作为基准，可以分别统计曝光后 1 天、3 天以及 7 天内产生的最终转化行为，从而得到不同成熟周期对应的监督标签：

$$\mathcal{D}_{1d}, \quad \mathcal{D}_{3d}, \quad \mathcal{D}_{7d} \tag{11.40}$$

其中，$\mathcal{D}_{1d}$ 主要反映短周期转化行为，具有较好的实时性；$\mathcal{D}_{3d}$ 能够覆盖更多中期反馈；而 $\mathcal{D}_{7d}$ 则能够较完整地统计长期购买、留存等行为，但相应的数据新鲜度也会有所下降。因此，不同时间窗口本质上反映了标签完整性（Completeness）与数据新鲜度（Freshness）之间的权衡关系。

为了同时利用不同时间尺度上的监督信号，工业界通常会构建多个延迟反馈物品塔（Delayed Item Tower），分别学习不同标签窗口对应的物品表示，如图11.8所示。例如，可以分别建立 1 天、3 天以及 7 天三个延迟反馈物品塔：

$$\mathbf{e}_{item}^{1d} = f_{1d}(\mathbf{x}_{item}), \qquad \mathbf{e}_{item}^{3d} = f_{3d}(\mathbf{x}_{item}), \qquad \mathbf{e}_{item}^{7d} = f_{7d}(\mathbf{x}_{item}) \tag{11.41}$$

对应地，各个塔分别完成不同时间窗口下的匹配预测：

$$s_{1d} = \mathbf{e}_{user}^{\top}\mathbf{e}_{item}^{1d}, \qquad s_{3d} = \mathbf{e}_{user}^{\top}\mathbf{e}_{item}^{3d}, \qquad s_{7d} = \mathbf{e}_{user}^{\top}\mathbf{e}_{item}^{7d} \tag{11.42}$$

需要注意的是，不同塔之间通常不会完全独立训练。为了避免模型规模快速增长，工业界一般共享 Embedding 层以及部分底层网络，仅在高层建立独立的塔进行表示学习。这样既能够充分利用各个任务之间的共享信息，又能够保证不同时间窗口对应的监督目标具有一定的独立表示能力。

除了模型结构之外，离线训练的另一个核心问题是训练样本的组织方式。由于不同标签窗口对应不同的数据成熟周期，因此训练数据通常按照不同日期进行切分。例如，在生成当天训练数据时，可以同时读取最近若干天曝光日志以及对应行为日志，根据曝光时间和标签窗口重新组织监督样本。假设曝光发生时间为 t_0，用户行为发生时间为 t_1，则对应窗口 W 内的监督标签可以表示为：

$$y_W = \begin{cases} 1, & t_1 - t_0 \le W, \\ 0, & \text{otherwise} \end{cases} \tag{11.43}$$

其中 W 可以分别对应 1 天、3 天或 7 天等不同统计窗口。

工业界通常采用统一的数据生成流程，根据不同窗口同时生成多种监督标签，因此最终形成的一条训练样本往往同时包含即时反馈标签以及多个延迟反馈标签，例如：

$$(y_{click}, y_{1d}, y_{3d}, y_{7d}) \tag{11.44}$$

随后，这些监督标签分别对应不同塔参与联合训练。假设即时反馈损失以及多个延迟反馈损失分别表示为

$$L_{click}, \quad L_{1d}, \quad L_{3d}, \quad L_{7d} \tag{11.45}$$

则整体训练目标通常可以写为

$$L = L_{click} + \lambda_1 L_{1d} + \lambda_2 L_{3d} + \lambda_3 L_{7d} \tag{11.46}$$

其中 λ_i 表示不同监督目标对应的权重系数，通常通过离线验证集或者线上 A/B 实验进行调优。

由于所有监督标签都来源于同一条训练样本，因此整个模型通常仍然只对应一个训练任务（Training Job），而不是分别维护多个独立模型。在训练过程中，用户塔始终利用所有监督目标共同更新参数，从而学习统一的用户兴趣表示；而不同延迟反馈物品塔则分别利用各自对应时间窗口的监督信号更新参数，学习不同时间尺度上的物品语义表示。这种设计既避免了多个训练任务之间的参数同步问题，也能够充分利用不同标签窗口之间的监督信息。

在线推理阶段，各个延迟反馈物品塔对应的物品 Embedding 通常都可以提前离线计算并写入 Embedding 服务，用户请求到来时仅需实时计算用户塔得到用户 Embedding，然后分别完成多个塔之间的向量匹配：

$$s_i = \mathbf{e}_{user}^{\top}\mathbf{e}_{item}^{i}, \qquad i \in \{click, 1d, 3d, 7d\} \tag{11.47}$$

最终，系统根据具体业务目标对多个 Tower 输出进行融合，例如采用加权求和或轻量级融合网络得到最终排序

分数：

$$s_{final} = \sum_i \alpha_i s_i \tag{11.48}$$

其中 α_i 表示不同业务目标对应的融合权重。

总体而言，相比实时训练更加关注模型更新速度，离线训练更加强调监督标签的完整性以及长期目标建模能力。通过构建不同标签成熟窗口对应的延迟反馈物品塔，并采用统一训练任务进行联合优化，工业推荐系统能够充分学习不同时间尺度上的用户反馈信息，从而显著提升长期价值相关目标（如购买率、支付率以及长期留存率）的预测能力。因此，目前大规模电商推荐、广告推荐以及本地生活推荐系统中，大多都会采用这种基于多标签窗口的离线延迟反馈建模方案。

11.2.3.3 实时训练与离线训练的比较

综上所述，针对延迟反馈（Delayed Feedback）的建模，目前工业界主要存在实时数据流训练（Online Incremental Training）与离线周期训练（Offline Periodic Training）两类典型方案。二者在训练样本的组织方式、标签成熟机制、模型更新频率以及适用业务目标等方面存在明显差异。近年来，越来越多工业级推荐系统并不是简单地将两类数据流混合到同一个训练任务中，而是采用模型级混合训练（Model-level Hybrid Training）架构，即针对不同反馈时效性与业务目标分别训练多个模型，并在在线服务阶段进行融合，从而同时兼顾模型的新鲜度（Freshness）与长期价值预测能力（Long-term Value Prediction）。

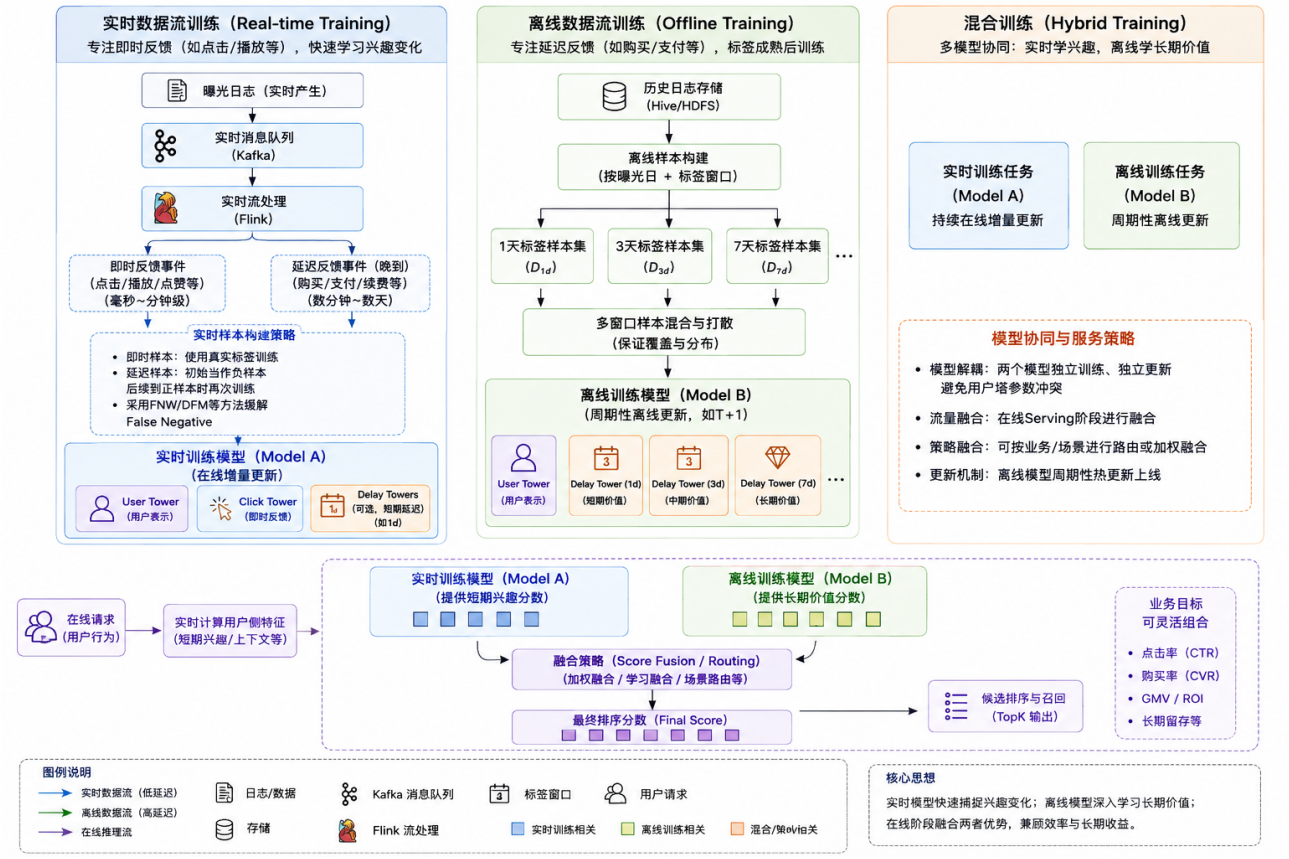

图 11.9: 延迟反馈多塔模型不同训练方式的整体架构对比。

如图 11.9所示，整个延迟反馈训练框架可以划分为三个部分。左侧为实时数据流训练（Real-time Training），中间为离线数据流训练（Offline Training），右侧则展示了工业界更加常见的混合训练（Hybrid Training）架构。需要注意的是，混合训练并非指将实时样本和离线样本直接合并到同一个训练任务中，而是针对不同监督目标分别构建独立模型，并在在线预测阶段进行联合使用。

图 11.9中左侧展示了实时训练的数据流流程及后续的训练过程。用户曝光日志首先进入 Kafka、Flink 等实时计算框架，当点击、播放、停留时长等即时反馈发生后，可以快速构建训练样本，并持续用于模型增量更新。由于标签产生延迟较短，模型能够快速捕获用户兴趣变化以及热点内容趋势。然而，由于购买、支付等延迟反馈行为尚未成熟，大量曝光样本在早期阶段可能被误认为负样本，从而产生 False Negative 问题。因此，实时训练通常需要结合标签修正（Label Correction）、样本重加权（Fake Negative Weighting）以及延迟反馈学习（Delayed Feedback Learning）等方法降低训练偏差。

图 11.9中间展示了离线训练的数据流组织方式以及模型训练过程。与实时训练不同，离线训练通常依赖 Hive、HDFS 等离线数仓系统，通过等待更长时间使监督标签充分成熟。例如，对于购买转化、复购以及长期价值等目标，可以构建曝光后 1 天、3 天甚至 7 天的标签窗口（Label Window），生成更加可靠的监督信号。由于训练样本中的延迟反馈已经充分回流，因此模型能够更加准确地学习长期用户价值，但由于采用小时级甚至天级周期更新，其对于最新用户兴趣变化的响应速度相对较慢。

图 11.9右侧展示了工业界常见的混合训练的架构。与前两种方式不同，混合训练通常不会将实时数据流与离线数据流直接输入同一个模型进行联合优化，而是分别训练多个面向不同目标的推荐模型。例如，实时模型（Real-time Model）主要基于实时行为流训练，负责优化点击、播放、互动等即时反馈目标；延迟反馈模型（Delayed Feedback Model）则基于离线成熟标签训练，重点优化购买、支付、长期留存以及用户生命周期价值等长期目标。在这种架构下，不同模型可以根据自身目标采用独立的数据流与更新策略。例如，实时模型可以通过分钟级甚至秒级的数据流持续更新，以保证对用户兴趣变化的快速响应；延迟反馈模型则采用小时级或天级周期训练，通过更加完整的转化标签不断校正模型对长期价值的预测能力。由于两个模型在训练过程中相互独立，因此能够避免不同时间尺度目标之间对用户表示产生梯度冲突，同时充分利用实时反馈和延迟反馈各自的优势。

从网络结构角度来看，实时模型和延迟反馈模型通常均可以采用多塔模型架构。每个模型内部均包含独立的用户塔以及对应目标的物品塔。其中，实时模型中的物品塔主要学习点击、播放等即时反馈相关表示，而延迟反馈模型中的物品塔则学习购买、支付等长期价值相关表示。不同模型之间通常不会共享训练过程中的用户塔参数，以避免不同监督目标之间的优化冲突，而是在在线推理阶段通过模型输出融合完成多目标推荐。

最终，如图 11.9下方所示，在在线推理阶段，推荐系统同时调用实时模型与延迟反馈模型完成预测。用户侧可以根据业务需求选择实时计算用户表示，或者分别获取不同模型对应的用户 Embedding；物品侧 Embedding 通常提前完成预计算并缓存至在线 Embedding 服务。随后，各模型分别计算对应目标的匹配分数，例如点击概率、观看时长预测、购买概率以及长期价值评分，并通过融合策略（Score Fusion）或多目标排序模型生成最终排序结果。

总体而言，延迟反馈建模的关键并不仅仅在于增加新的延迟反馈物品塔，而是在于根据不同反馈时间尺度设计合理的数据流架构与模型训练策略。实时训练能够提供快速的兴趣响应能力，离线训练能够利用成熟标签提升长期价值预测准确性，而模型级混合训练则进一步结合二者优势，避免多目标联合优化带来的训练冲突。因此，目前工业推荐系统通常采用实时模型与离线延迟反馈模型协同工作的架构，并广泛应用于电商推荐、广告推荐、本地生活推荐等需要同时优化短期体验与长期业务价值的场景。

11.2.4 带交叉特征的多塔模型

除了针对不同物品类型或延迟反馈场景建立多个塔之外，另一类广泛应用于工业级粗排模型的多塔结构，则是在传统双塔模型之外增加专门用于建模用户与物品复杂交互关系的**交叉特征塔（Cross Interaction Tower）**。该类模型目前已经成为工业界粗排模型中最常见的网络结构之一，其核心思想如下：

定义 11.6 (交叉特征塔)

交叉特征塔是指在传统双塔模型之外，额外增加的用于显式建模用户与物品之间重要组合特征的独立网络结构。该塔通常接收用户与物品两侧的关键组合特征作为输入，通过轻量级神经网络学习显式交互关系，从而增强模型对用户与物品之间复杂交互关系的建模能力。

传统 DSSM 模型采用双塔结构分别学习用户 Embedding 与物品 Embedding，两侧网络完全独立，仅在网络

末端通过向量内积计算匹配分数：

$$s = \mathbf{e}_{user}^{\top} \mathbf{e}_{item} \tag{11.49}$$

这种结构最大的优势在于在线计算效率极高。由于用户塔与物品塔相互独立，在线阶段仅需计算一次用户 Embedding，而物品 Embedding 可以提前离线预计算，因此整个在线排序过程仅需要完成一次 Embedding 内积即可得到匹配分数。

然而，双塔模型的这一独立编码方式也带来了天然的局限性。由于用户特征与物品特征在编码过程中始终彼此独立，双塔模型只能依赖 Embedding 空间中的隐式语义相似性完成匹配，而难以直接学习具有明确业务含义的显式交互关系。例如，不同年龄段用户对于不同内容类别的偏好、不同消费能力用户对于不同价格区间商品的购买倾向，以及不同地域用户对于本地生活内容的兴趣差异等，都属于用户与物品之间典型的组合特征，仅依赖 Embedding 内积通常难以充分表达。

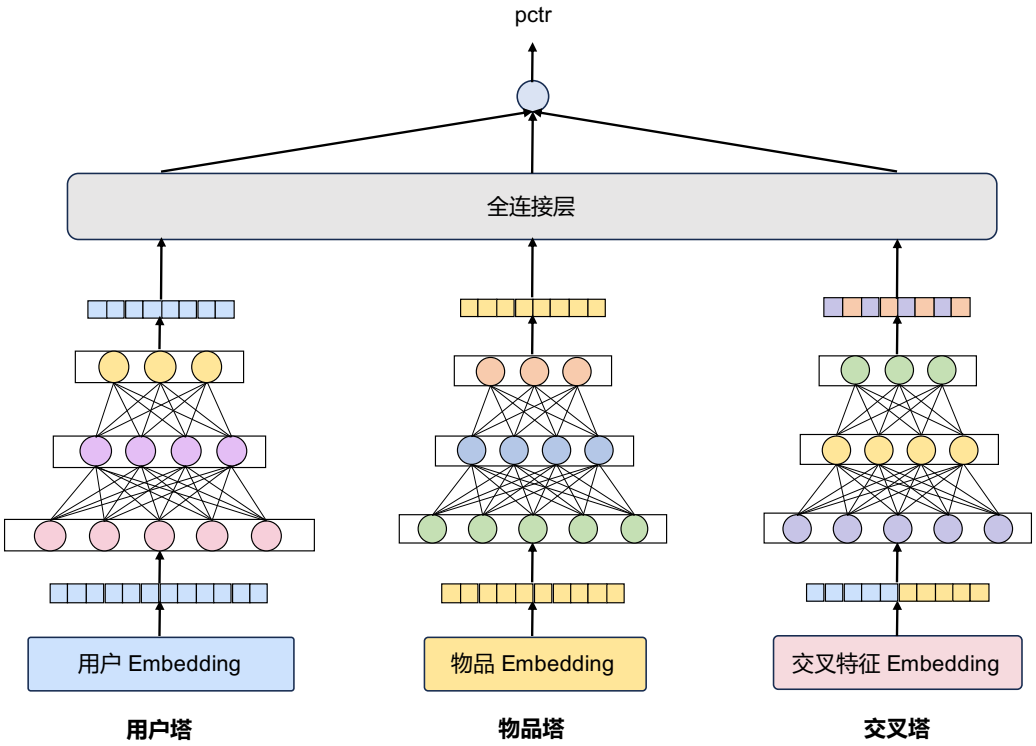

图 11.10: 带交叉特征的多塔模型整体结构示意图。

因此，近年来工业界通常会在传统双塔模型之外增加一个或多个交叉特征塔（Cross Interaction Tower），专门用于学习用户与物品之间的显式交互关系。整个模型一般由用户塔（User Tower）、物品塔（Item Tower）以及交叉特征塔（Cross Tower）共同组成，如图 11.10所示。三个 Tower 通常承担着不同的建模职责，因此其输入特征以及网络结构也存在明显差异。

用户塔主要负责学习用户长期兴趣表示，其输入通常包括用户画像、用户 ID、设备信息、用户位置、用户性别、历史行为统计特征以及长短期兴趣 Embedding 等用户侧特征，通过若干层全连接网络得到用户 Embedding：

$$\mathbf{e}_{user} = f_{user}(\mathbf{x}_{user}) \tag{11.50}$$

而物品塔主要负责学习物品的语义表示，其输入通常包括物品 ID、作者 ID、作品上传时间、类目、标签、文本 Embedding、图像 Embedding 以及视频 Embedding 等物品侧特征，并输出统一的物品 Embedding：

$$\mathbf{e}_{item} = f_{item}(\mathbf{x}_{item}) \tag{11.51}$$

与前两个塔不同，交叉特征塔并不学习用户或物品的独立表示，而是直接接收用户与物品两侧的重要组合特征作为输入，例如用户 ID 与物品 ID、用户年龄与物品类目、用户地域与商家位置、设备类型与内容类型等具

有明确业务意义的组合特征，通过轻量级神经网络学习显式交互关系：

$$\mathbf{h}_{cross} = f_{cross}(\mathbf{x}_{user}, \mathbf{x}_{item}) \tag{11.52}$$

其中，$f_{cross}(\cdot)$ 可以采用 FM、DeepFM、DCN、MLP 等不同形式实现。当交叉特征数量较少时，FM 通常能够以较低计算成本建模二阶交互；而当需要学习更加复杂的非线性交互关系时，则可以采用 DCN 或浅层 MLP 进行建模。

三个塔分别完成表示学习之后，通常将用户 Embedding、物品 Embedding 以及特征交叉 Embedding 进行拼接，再输入后续融合网络：

$$\mathbf{h} = \text{Concat}(\mathbf{e}_{user}, \mathbf{e}_{item}, \mathbf{h}_{cross}) \tag{11.53}$$

最终经过若干层全连接网络得到最终预测分数：

$$\hat{y} = \text{MLP}(\mathbf{h}) \tag{11.54}$$

需要指出的是，三个塔虽然共同组成了整个粗排模型，但在网络规模设计上通常存在较大差异。由于用户塔和物品塔主要负责学习高层语义表示，因此通常采用相对较深的网络结构，例如 2~4 层全连接网络，每层隐藏维度一般为 256~1024 维。而交叉特征塔主要负责学习少量重要组合特征，因此网络通常设计得更加轻量，仅包含 1~2 层全连接网络，甚至直接采用 FM 或 Cross Network 进行建模，从而避免复杂交互网络带来的在线计算开销。

此外，三个塔所使用的输入特征也具有不同特点。用户塔主要使用稳定的用户画像和历史兴趣特征，物品塔主要使用物品内容与属性特征，而交叉特征塔则更加关注少量具有较强业务意义的组合特征。在工业实践中，交叉特征通常会经过严格的人工筛选或自动特征选择，仅保留对预测效果贡献最大的几十到数百个交叉特征，以避免组合特征数量爆炸导致模型复杂度急剧增加。

在线推理阶段，三个塔的计算方式同样存在明显区别。用户塔通常在每次用户请求到达时执行一次前向推理，生成当前用户的 Embedding 表示；物品塔由于仅依赖物品自身特征，因此可以提前完成离线 Embedding 预计算，并缓存至在线 Embedding 服务中，从而避免重复计算。而交叉特征塔由于需要同时访问用户特征与物品特征，因此只能在在线阶段实时计算。与此同时，三个塔输出的 Embedding 在完成拼接之后，后续融合网络同样需要实时完成一次前向推理。因此，在工业级推荐系统中，交叉特征塔以及最终融合网络通常都会采用较为轻量的网络结构。例如，交叉特征塔一般仅包含 1~2 层隐藏层，而融合网络通常也仅采用 1~2 层全连接层，以保证整个粗排模型能够满足毫秒级在线推理时延要求。

总体来看，相比传统 DSSM 模型仅依赖 Embedding 相似度完成语义匹配，增加交叉特征塔之后，模型能够同时兼顾隐式语义匹配能力与显式交互建模能力，从而有效提升粗排阶段对于复杂用户兴趣和高阶交互关系的建模效果。目前，“用户塔 + 物品塔 + 交叉特征塔”已经成为工业界粗排模型最主流的网络架构之一，而交叉特征塔也可以根据具体业务需求灵活替换为 FM、DeepFM、DCN、Cross Network 或其他轻量级交互模型，在保证在线推理效率的同时进一步提升推荐效果。

11.3 粗排蒸馏模型

工业级推荐系统中的粗排模型需要在极短的在线时延约束下，对数千甚至上万个候选物品完成快速打分，因此模型通常采用双塔（Two-Tower）、多塔（Multi-Tower）等轻量级网络结构，以满足在线推理效率要求。然而，这类模型在兼顾效率的同时，也不可避免地牺牲了一部分模型表达能力。具体问题如下：

笔记 由于粗排模型需要在严格的在线时延约束下完成大规模候选集的快速排序，因此通常采用双塔、多塔等轻量级网络结构。这类模型虽然具有较高的推理效率，但整体表达能力有限，难以充分建模用户与物品之间复杂的高阶交互关系，从而限制了粗排模型的排序精度。

针对上述问题，工业界主要存在两类优化思路。第一类方法是在保持粗排模型整体架构不变的基础上，通过增强模型的表达能力提升排序效果。例如，上一章节介绍的 Cross Tower 方法，便是在传统双塔或多塔模型之外增加交叉特征建模的 DNN 分支，使模型能够显式学习用户与物品之间的高阶交互关系，从而弥补双塔模型交互能力不足的问题。第二类方法则是在保持在线推理结构基本不变的前提下，引入**知识蒸馏（Knowledge Distillation）**

技术，将表达能力更强的复杂模型所学习到的知识迁移至轻量级粗排模型中。相比直接增加模型规模，知识蒸馏几乎不会带来额外的在线推理开销，却能够充分利用教师模型所蕴含的高阶特征表示和排序知识，因此已经成为近年来工业界优化粗排模型的重要手段之一。

知识蒸馏（Knowledge Distillation）的基本思想如下：

定义 11.7 (知识蒸馏)

知识蒸馏旨在构建一个性能更优、表达能力更强的教师模型（Teacher Model），利用其输出结果或中间表示作为额外监督信号，引导轻量级学生模型（Student Model）的训练。

这样，学生模型不仅学习真实标签（Ground Truth），还能够进一步学习教师模型所蕴含的类别关系、样本相似性以及更加丰富的特征表示，从而在模型结构几乎保持不变的情况下获得更好的预测性能。

近年来，知识蒸馏已经广泛应用于推荐系统、广告排序、搜索排序等工业场景，并逐渐成为粗排模型优化中的标准技术方案。根据教师模型构建方式、知识传递形式以及训练流程的不同，目前工业界已经提出了多种蒸馏方法。其中，有的方法采用预训练教师模型指导学生模型学习，有的方法则采用教师模型与学生模型同步训练，还有一些工作进一步结合中间层表示蒸馏、多教师蒸馏以及自蒸馏（Self-distillation）等技术，以进一步提升轻量模型的表达能力。

本章节主要介绍工业界粗排模型中几种具有代表性的蒸馏方法，包括经典 Teacher-Student 蒸馏框架、Rocket Launching 模型以及 PFD 模型，并重点分析它们的模型结构、训练方式以及知识迁移机制。

11.3.1 Teacher-Student 模型

Teacher-Student（教师-学生）框架是目前知识蒸馏（Knowledge Distillation）最经典也是应用最广泛的基本框架。该框架最早由 Hinton 等人在 2015 年发表于论文《Distilling the Knowledge in a Neural Network》[3]。其核心思想是利用一个表达能力更强的教师模型（Teacher Model）指导轻量级学生模型（Student Model）的训练，使学生模型不仅学习真实标签（Ground Truth），还能够学习教师模型所蕴含的类别关系、特征表示以及高阶交互信息，从而在保持在线推理效率基本不变的情况下获得更好的预测性能。

假设教师模型表示为 $f_T(\mathbf{x})$，学生模型表示为 $f_S(\mathbf{x})$，传统 Teacher-Student 框架的优化目标通常可以表示为

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{task}(y, f_S(\mathbf{x})) + \lambda\mathcal{L}_{KD}(f_T(\mathbf{x}), f_S(\mathbf{x})), \tag{11.55}$$

其中，$\mathcal{L}_{task}$ 表示学生模型在真实标签上的监督损失，$\mathcal{L}_{KD}$ 表示知识蒸馏损失，λ 用于平衡监督学习与知识蒸馏的重要性。根据知识传递方式的不同，工业界常见的 Teacher-Student 框架主要可以分为显式知识蒸馏（Explicit Knowledge Distillation）和隐式知识蒸馏（Implicit Knowledge Distillation）两大类。

11.3.1.1 显式知识蒸馏

显式知识蒸馏是最经典的 Teacher-Student 训练方式，其特点是在训练过程中显式构建蒸馏损失（Distillation Loss），直接约束教师模型与学生模型之间的输出或中间表示一致，从而实现知识迁移。

11.3.1.1.1 输出层蒸馏 输出层蒸馏（Output Distillation）也是 Hinton 提出的经典 Knowledge Distillation（KD）方法。其基本思想是在真实标签之外，再利用教师模型输出的 Soft Label 作为额外监督信号。为了保留类别之间更加丰富的关系，KD 通常采用带温度参数（Temperature）的 Softmax：

$$p_i^{(T)} = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \tag{11.56}$$

其中，T 表示温度参数。当 $T > 1$ 时，Softmax 输出更加平滑，学生模型能够学习教师模型对于不同类别之间的相对偏好，而不仅仅是最终预测类别。

输出层蒸馏方法对应的蒸馏损失通常采用交叉熵：

$$\mathcal{L}_{KD} = H\left(p_T^{teacher}, p_T^{student}\right) \tag{11.57}$$

这样，整个模型的最终训练目标为

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{task} + \lambda\mathcal{L}_{KD} \tag{11.58}$$

需要注意的是，输出层蒸馏这种方法实现简单，目前仍然是工业界最常见的知识蒸馏方式之一。

11.3.1.1.2 中间层表示蒸馏 除了最终输出之外，教师模型中间层所学习到的隐藏表示（Hidden Representation）同样包含大量有价值的信息。因此，中间层表示蒸馏进一步约束教师模型与学生模型对应层的特征表示保持一致。

假设教师模型第 l 层隐藏表示为 $\mathbf{h}_T^{(l)}$，学生模型对应层表示为 $\mathbf{h}_S^{(l)}$，则常见蒸馏损失可以表示为

$$\mathcal{L}_{feature} = \left\|\mathbf{h}_T^{(l)} - \mathbf{h}_S^{(l)}\right\|_2^2 \tag{11.59}$$

最终整个粗排蒸馏模型目标函数可进一步表示为

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_1\mathcal{L}_{KD} + \lambda_2\mathcal{L}_{feature} \tag{11.60}$$

相比起只学习最终的输出，中间层蒸馏能够帮助学生模型学习更加丰富的特征表示，因此在工业级推荐系统中也被广泛应用。

11.3.1.2 隐式知识蒸馏

与显式知识蒸馏不同，隐式知识蒸馏（Implicit Knowledge Distillation）通常不会额外设计蒸馏损失函数，而是通过教师模型与学生模型联合训练（Joint Training）、共享部分网络结构以及共享梯度传播路径，使学生模型在训练过程中隐式学习教师模型所蕴含的知识。

近年来，隐式知识蒸馏在工业推荐系统中的应用越来越广泛，其主要原因在于推荐模型通常具有大量共享 Embedding 以及复杂的多塔结构，相比额外设计蒸馏 Loss，共享网络结构往往更加简单高效。根据共享位置的不同，工业界常见的隐式知识蒸馏主要包括以下三种方式。

11.3.1.2.1 输入 Embedding 共享 第一种方法是在输入层共享 Embedding，如图 11.11(a) 所示。教师模型通常采用复杂的全连接网络，输入既包括用户侧特征、物品侧特征，也包括用户与物品之间的交叉特征；学生模型仍然采用轻量级双塔结构，仅使用用户侧和物品侧特征。虽然两个模型的网络结构不同，但二者共享底层 Embedding 参数，因此 Embedding 同时受到教师模型与学生模型梯度更新，用公式可以表示为：

$$\mathbf{E} \leftarrow \mathbf{E} - \eta\left(\frac{\partial\mathcal{L}_{teacher}}{\partial\mathbf{E}} + \frac{\partial\mathcal{L}_{student}}{\partial\mathbf{E}}\right) \tag{11.61}$$

其中，$\mathbf{E}$ 表示共享 Embedding 参数。由于教师模型能够学习更加复杂的高阶交互关系，因此 Embedding 参数也会学习到更加丰富的语义表示，从而间接提升学生模型性能。

11.3.1.2.2 中间层共享 第二种方式是在网络中间层共享隐藏表示，如图 11.11(b) 所示。假设共享隐藏表示为 $\mathbf{h}$，则教师模型与学生模型共同优化该隐藏表示：

$$\mathbf{h} \leftarrow \mathbf{h} - \eta\left(\frac{\partial\mathcal{L}_{teacher}}{\partial\mathbf{h}} + \frac{\partial\mathcal{L}_{student}}{\partial\mathbf{h}}\right) \tag{11.62}$$

相比起输入层共享，中间层已经完成一定程度的非线性特征交互，因此学生模型能够学习到教师模型更加深层次的表示能力。

11.3.1.2.3 顶层 Embedding 联合预测 第三种方式是在顶层 Embedding 进行联合训练，如图 11.11(c) 所示。教师模型产生复杂交互后的 Embedding，学生模型产生双塔的顶层 Embedding，二者共同参与最终预测：

$$\mathbf{h}_{fusion} = [\mathbf{u}_S, \mathbf{i}_S, \mathbf{u}_S \odot \mathbf{i}_S, \mathbf{h}_T] \tag{11.63}$$

其中，$\mathbf{u}_S$ 和 $\mathbf{i}_S$ 分别表示学生模型用户侧和物品侧双塔的顶层 Embedding，$\odot$ 表示逐元素乘积，$\mathbf{h}_T$ 表示教师模型顶层表示。最终的预测输出为：

$$\hat{y} = \sigma(\text{MLP}(\mathbf{h}_{fusion})) \tag{11.64}$$

在模型的训练过程中，学生模型顶层 Embedding 不断参与最终预测，因此能够持续受到教师模型高阶交互信息的影响，实现隐式知识迁移。而在线推理阶段只保留学生模型进行推理，教师模型也会完全移除，因此并不会增加任何在线计算开销。

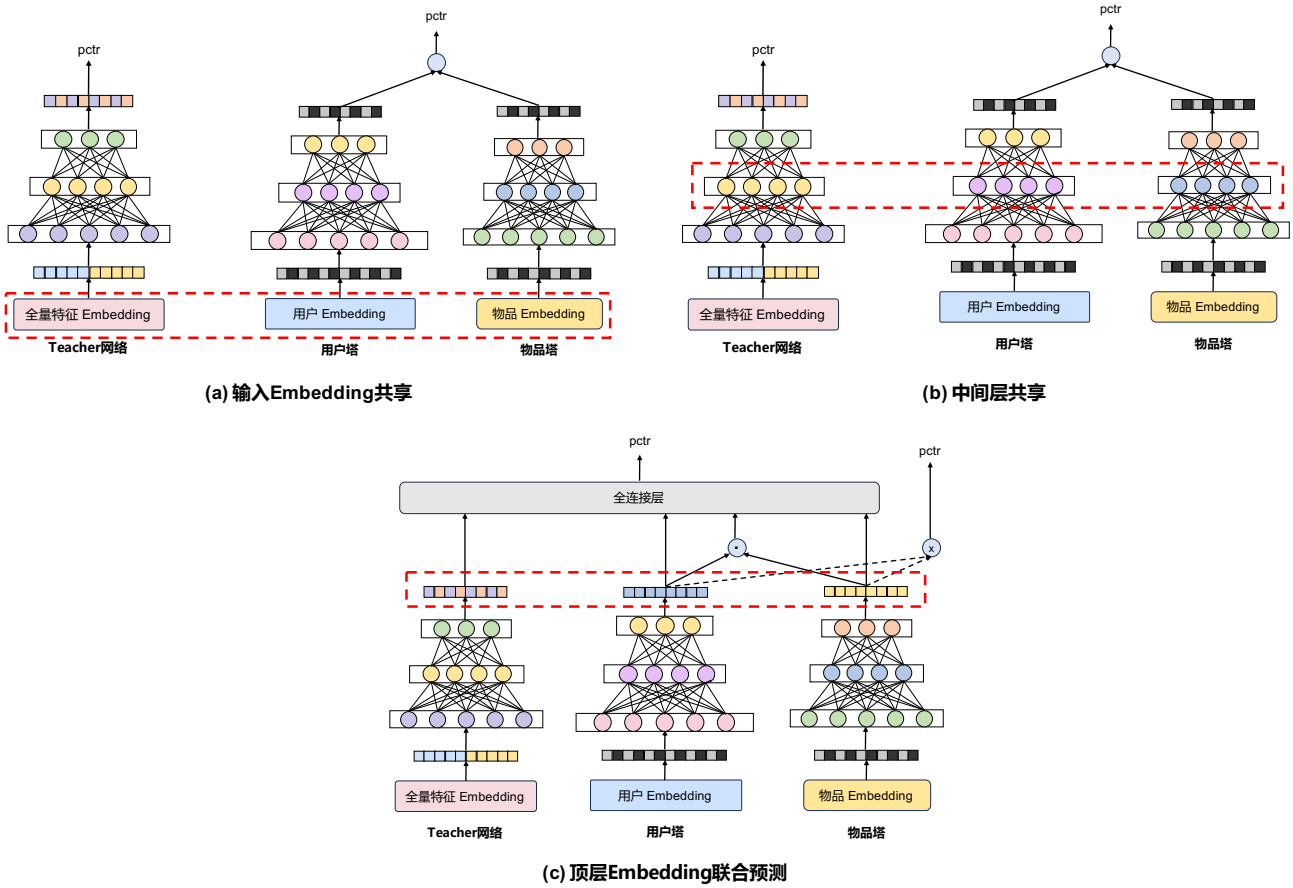

图 11.11: 隐式知识蒸馏的 Teacher-Student 联合训练示意图。

总体而言，显式知识蒸馏通过设计蒸馏损失函数直接约束教师模型与学生模型的一致性，而隐式知识蒸馏则通过联合训练、参数共享以及共享表示实现知识迁移。两类方法本质上都是希望利用复杂教师模型提升轻量级学生模型的表达能力，只是在知识传递方式上有所不同。后续介绍的 Rocket Launching 以及 PFD 模型，均可以看作是在 Teacher-Student 框架基础上的进一步改进，其中 Rocket Launching 重点改进了训练流程，而 PFD 则进一步引入了训练阶段可获得但在线不可用的特权特征，实现更高效的知识迁移。

11.3.2 Rocket Launching 模型

Rocket Launching 模型由阿里巴巴于 2018 年提出，发表于 AAAI 会议论文《Rocket Launching: A Universal and Efficient Framework for Training Well-Performing Light Net》[16]。该方法针对工业场景下轻量级模型性能不足的问题，提出了一种共享参数、同步训练（Simultaneous Training）的知识蒸馏框架，使学生模型能够在训练阶段持续学习教师模型的知识，从而显著提升轻量模型的预测能力。

与传统 Teacher-Student 蒸馏不同，Rocket Launching 最大的特点并不是提出新的蒸馏损失，而是在训练框架上进行了改进。整个网络由轻量级学生网络（Light Net）与高容量教师网络（Booster Net）组成，如图 11.12所示。两者共享底层特征提取网络，仅在高层分别构建各自的预测分支。其中，学生网络作为最终在线部署模型，

而教师网络仅参与训练过程，用于提供额外的监督信息。

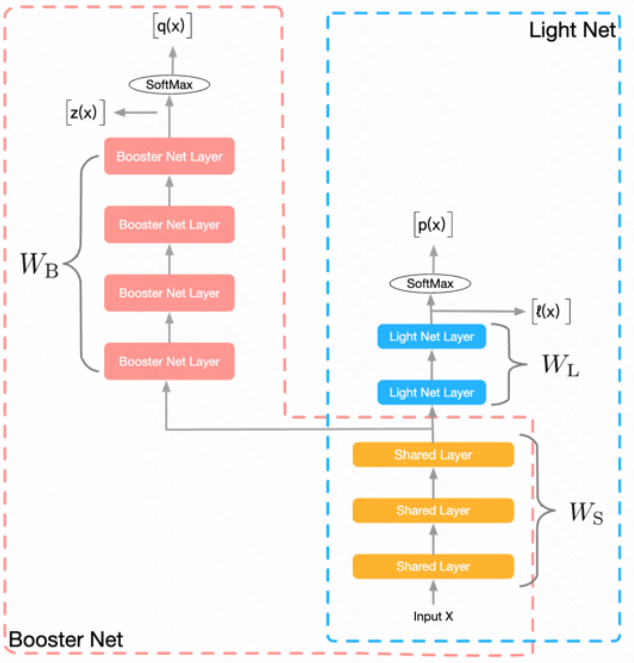

图 11.12: Rocket Launching 模型整体结构示意图。

假设输入样本为 x，真实标签为 y。学生网络输出的 Logits 记为 $l(x)$，对应预测概率为 $p(x) = \text{softmax}(l(x))$；教师网络输出 Logits 记为 $z(x)$，预测概率为 $q(x) = \text{softmax}(z(x))$。其中，共享参数记为 W_S，学生网络特有参数记为 W_L，教师网络特有参数记为 W_B。

Rocket Launching 采用联合优化方式同时训练教师网络与学生网络。训练过程中，两者均利用真实标签进行监督学习，同时教师网络还会通过 Hint Loss 向学生网络传递知识。因此，其整体优化目标可以表示为

$$\mathcal{L} = H(y, p(x)) + H(y, q(x)) + \lambda L_{\text{hint}} \tag{11.65}$$

其中，$H(\cdot)$ 表示交叉熵损失（Cross Entropy），前两项分别对应学生网络和教师网络的监督损失，而 L_{hint} 表示蒸馏损失（Hint Loss），用于约束学生网络学习教师网络所蕴含的知识，λ 用于平衡监督学习与知识蒸馏之间的重要性。

Rocket Launching 论文进一步比较了多种不同形式的 Hint Loss，其核心思想均是希望学生网络能够尽可能逼近教师网络的输出表示。目前主要包括以下三种方式。第一种是 Softmax 输出均方误差（Softmax MSE），即约束教师网络与学生网络最终预测概率之间的均方误差：

$$L_{\text{MSE}} = \|p(x) - q(x)\|_2^2 \tag{11.66}$$

该方法直接匹配两个模型最终输出的概率分布，实现较为简单，但由于 Softmax 会压缩 Logits，当某些类别概率接近于零时，其梯度也会迅速减小，因此容易出现梯度消失的问题。第二种方法是 Logits 均方误差（Logits MSE），这种方法不经过 Softmax，而是直接约束教师网络与学生网络 Softmax 之前的 Logits 输出，即：

$$L_{\text{Mimic}} = \|l(x) - z(x)\|_2^2 \tag{11.67}$$

相比起 Softmax 输出，Logits 保留了模型对于所有类别更加丰富的相对关系，因此直接匹配 Logits 不仅能够避免梯度消失，还能够保留更多类别之间的结构信息，因此也是 Rocket Launching 最终采用的 Hint Loss 形式。最后一种方法是 Knowledge Distillation（KD），即经典知识蒸馏方法。这种方法采用带温度系数的 Softmax 概率作为监督信号，其蒸馏损失可以表示为：

$$L_{\text{KD}} = H\left(p_T(x), q_T(x)\right) \tag{11.68}$$

其中，$p_T(x)$ 和 $q_T(x)$ 分别表示经过温度参数 T 缩放后的 Softmax 输出。当温度较高时，类别概率分布更加平滑，学生网络能够学习教师模型对不同类别之间的相对偏好，因此该方法也是目前知识蒸馏中最经典的实现方式。

Rocket Launching 论文对上述三种 Hint Loss 进行了实验比较。实验结果表明，直接采用 Logits 均方误差能够取得最佳性能。这是因为 Softmax 概率经过归一化后容易造成梯度衰减，而 KD 虽然能够获得更加平滑的类别分布，但仍然需要经过 Softmax 变换。相比之下，Logits MSE 直接作用于 Softmax 之前的输出，不仅优化过程更加稳定，也能够保留教师模型更多隐含的类别关系，因此最终被 Rocket Launching 作为默认的知识蒸馏方式。

可以看出，Rocket Launching 模型并没有仅利用教师模型预测出的类别作为监督，而是直接约束两者 Logits 之间的距离。相比于仅学习最终分类结果，Logits 包含了模型对于所有类别的相对偏好，因此能够保留更多类别之间的结构信息，使学生模型学习到更加丰富的知识表示。

Rocket Launching 与传统 Teacher-Student 方法相比，主要具有以下几个特点。首先，Rocket Launching 采用了参数共享（Parameter Sharing）机制。教师网络与学生网络共享底层 Embedding 层以及基础特征提取网络，仅在高层构建各自独立的预测分支。这种设计不仅减少了整体训练参数规模，同时也使学生网络能够直接受益于教师网络学习到的底层特征表示，提高模型收敛速度与泛化能力。类似的参数共享思想目前也广泛应用于工业推荐系统中的多任务学习（Multi-task Learning）以及多塔模型中。

其次，Rocket Launching 采用同步训练（Simultaneous Training）策略，而不是传统蒸馏方法中“先训练 Teacher，再固定 Teacher 训练 Student”的两阶段流程。训练过程中，教师网络和学生网络共同学习真实标签，同时教师网络持续为学生网络提供知识指导。相比传统离线蒸馏方式，同步训练无需额外训练一个教师模型，大幅降低了训练成本，也更加符合工业推荐系统频繁更新模型的需求。

最后，为了避免学生网络反向影响教师网络的学习，论文进一步提出了梯度阻断（Gradient Block）机制。由于 Hint Loss 同时连接教师网络和学生网络，如果直接反向传播，教师网络不仅受到真实标签监督，还会受到学生网络蒸馏损失的约束，从而降低教师模型自身的表达能力。为了解决这一问题，Rocket Launching 在 Hint Loss 反向传播过程中阻断教师网络分支的梯度，仅利用教师网络当前输出作为固定监督目标指导学生网络学习，而教师网络仍然仅依赖真实标签进行优化。这样既保证了教师模型能够持续学习最优表示，又避免学生网络对教师网络产生负向影响。

总体而言，Rocket Launching 并没有设计复杂的蒸馏网络结构，而是在传统知识蒸馏框架基础上，通过参数共享、同步训练以及 Gradient Block 三项简单而有效的改进，使轻量级模型能够持续学习高容量模型的知识，在几乎不增加在线推理成本的情况下显著提升预测效果。因此，该方法也成为工业界粗排模型知识蒸馏的代表性工作之一，并为后续 PFD 等蒸馏模型奠定了基础。

11.3.3 PFD 模型

PFD（Privileged Feature Distillation）模型源自阿里巴巴发表于 KDD 2020 论文《Privileged Features Distillation at Taobao Recommendations》[12]。该方法针对工业推荐系统粗排模型在线特征受限的问题，提出了一种基于特权特征（Privileged Features）的知识蒸馏方法，使轻量级粗排模型能够在训练阶段学习更多信息，而在线推理阶段仍保持原有的计算效率。

定义 11.8

PFD（Privileged Feature Distillation）的核心思想是在训练阶段构建一个使用特权特征（Privileged Features）的教师模型，通过知识蒸馏将这些额外特征所蕴含的信息迁移至仅使用在线特征的学生模型，从而在不增加在线推理开销的情况下提升模型性能。

需要注意的是，PFD 模型并不是传统知识蒸馏的简单应用，而是来源于 Learning Using Privileged Information（LUPI）的思想。LUPI 认为，在训练阶段往往能够获得一些在线推理阶段无法使用的额外信息，例如人工标注信息、未来行为、复杂统计特征、用户后验反馈信息等。如果能够利用这些额外的信息来训练教师模型，再将教师模型的知识迁移至学生模型，则能够进一步提升学生模型的泛化能力。

然而，PFD 模型的论文指出，直接采用 LUPI 仍然存在一定局限性。传统 LUPI 仅利用特权特征训练教师

模型，即教师模型输入为 $\mathbf{x}_{privileged}$，学生模型输入为在线可用特征 $\mathbf{x}$。由于特权特征通常只是全部特征的一部分，仅依赖这些特征训练得到的教师模型未必比学生模型更强，甚至可能产生误导性的预测结果。例如，在电商 CVR 预测中，用户浏览商品详情页停留时间属于典型的特权特征。虽然停留时间通常能够反映用户兴趣，但如果教师模型仅依据停留时间进行预测，而忽略商品价格等在线可获得的重要特征，则可能错误地认为所有停留时间较长的商品都具有较高的购买概率。因此，仅利用特权特征训练教师模型并不能充分发挥知识蒸馏的作用。

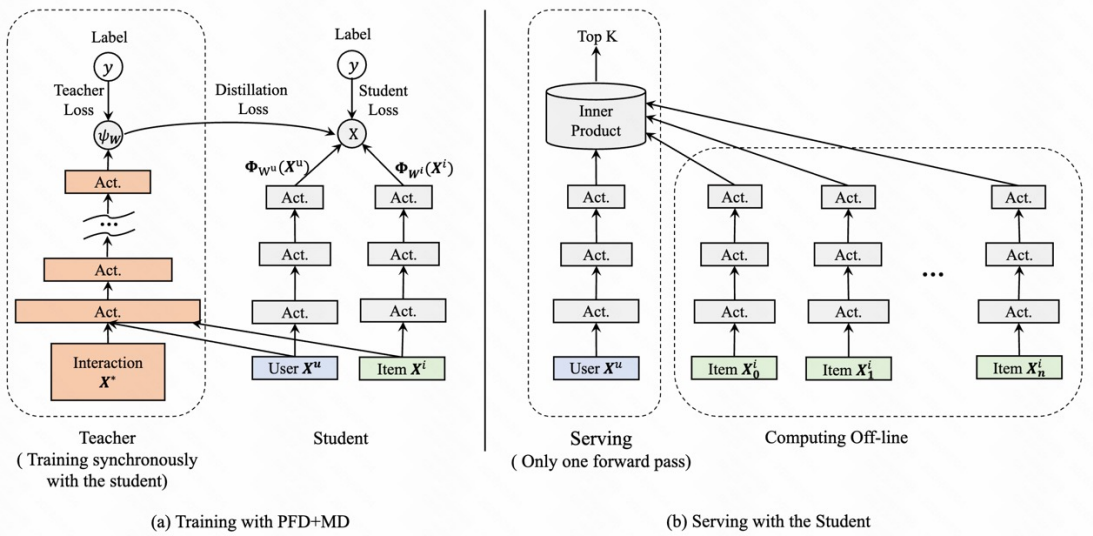

图 11.13: PFD 模型整体结构示意图。

针对这一问题，PFD 进一步改进了 LUPI 框架。与传统 LUPI 不同，PFD 中的教师模型不仅使用特权特征，同时也使用学生模型所采用的全部在线特征。因此，教师模型输入由原来的 $\mathbf{x}_{privileged}$ 扩展为 $(\mathbf{x}, \mathbf{x}_{privileged})$，学生模型仍然仅使用在线可获得特征 $\mathbf{x}$，整体结构如图 11.13所示。假设教师模型表示为 $f_T(\mathbf{x}, \mathbf{x}_{privileged})$，学生模型表示为 $f_S(\mathbf{x})$，则 PFD 的目标函数可以表示为

$$\mathcal{L} = (1-\lambda)\mathcal{L}_S\left(y, f_S(\mathbf{x})\right) + \lambda\mathcal{L}_{KD}\left(f_T(\mathbf{x}, \mathbf{x}_{privileged}), f_S(\mathbf{x})\right) + \mathcal{L}_T\left(y, f_T(\mathbf{x}, \mathbf{x}_{privileged})\right), \tag{11.69}$$

其中，$\mathcal{L}_S$ 表示学生模型监督损失，$\mathcal{L}_T$ 表示教师模型监督损失，$\mathcal{L}_{KD}$ 表示知识蒸馏损失，λ 用于平衡监督学习与蒸馏学习。

与传统 Teacher-Student 蒸馏相比，PFD 最大的区别在于教师模型能够同时利用在线特征与特权特征，因此具有更强的表达能力。相比传统 LUPI 仅依赖特权特征构建教师模型，PFD 中的教师模型能够综合利用全部可获得的信息进行预测，从而产生更加准确、更加稳定的软标签（Soft Label），进一步提高学生模型的学习效果。

图 11.14展示了 PFD 与传统知识蒸馏方法之间的区别。传统知识蒸馏主要依赖教师模型输出的概率分布或中间表示进行知识迁移，而 PFD 则进一步利用训练阶段可获得、在线阶段不可获得的特权特征来增强教师模型，使教师模型本身具有更强的预测能力，因此能够向学生模型传递更加丰富的知识。

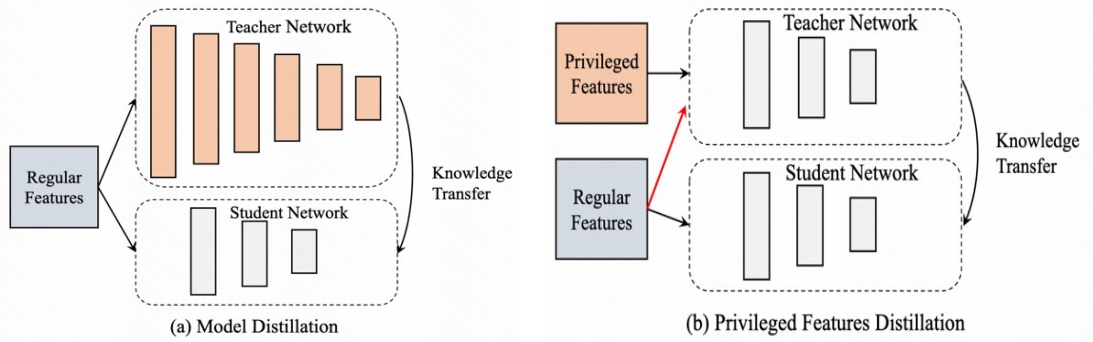

图 11.14: PFD 模型与传统知识蒸馏模型的区别示意图。

为了降低工业环境中的训练成本，论文进一步采用了同步训练（Synchronous Training）策略，而不是传统蒸

馏中先训练教师模型、再训练学生模型的两阶段流程。训练过程中，教师模型与学生模型同时进行优化，并利用教师模型当前输出持续指导学生模型学习。考虑到训练初期教师模型尚未充分收敛，论文进一步采用 Warm-up 策略，在训练初期仅优化教师模型与学生模型各自的监督损失，当教师模型达到一定训练阶段后，再逐步引入蒸馏损失，从而避免早期教师模型产生不稳定监督信号。

与此同时，PFD 还采用了参数共享（Parameter Sharing）机制。教师模型与学生模型共享 Embedding 层以及用户行为编码模块，仅在高层网络保留各自独立的预测分支。由于 Embedding 参数通常占据推荐模型的大部分存储空间，而用户行为序列编码也是训练过程中计算开销最大的模块，因此共享这些公共部分不仅能够显著降低参数服务器之间的通信量，还能够减少大量重复计算，使训练成本相比单独训练学生模型仅增加少量额外开销，更适合工业级在线学习系统。

值得注意的是，PFD 模型中的特权特征并不局限于某一种特征，而是指所有训练阶段可获得、在线推理阶段不可获得或计算成本过高的特征。例如，在粗排 CTR 预测中，用户过去 24 小时在某一商品类目上的点击次数、用户与商品之间复杂的交叉统计特征等。当计算资源紧张时，由于需要针对每一个候选商品实时计算，因此难以应用于在线粗排模型。除此之外，在 CVR 预测中，用户浏览详情页后的停留时长、是否查看评论、是否咨询客服等未来行为特征在先验的推理阶段也是无法获取的，只有在用户点击之后才能获得，因此这些用户后验的行为反馈特征同样属于典型的特权特征。

从这个角度来看，PFD 模型与前面介绍的带交叉特征多塔模型具有一定的相似性，两者都会利用更加复杂的交叉特征提升模型表达能力。然而，两者最大的区别在于这些复杂特征的使用方式不同。交叉特征多塔模型需要在训练阶段和在线推理阶段同时计算交叉特征以及执行后续所有特征显示交叉的全连接层网络，因此会显著增加在线计算开销；而 PFD 模型只在训练阶段利用特权特征构建教师模型，在线推理阶段仅保留学生模型进行预测，因此既能够充分利用复杂特征提升模型性能，又不会增加任何额外的在线推理延迟。这也是 PFD 模型能够广泛应用于工业级粗排模型优化的重要原因。

11.4 粗排 DNN 模型

前面的几节主要介绍了工业级推荐系统中常见的双塔模型（Two-Tower）、多塔模型（Multi-Tower）以及粗排蒸馏模型（Knowledge Distillation）。这些模型均围绕如何在保证在线推理效率的前提下提升粗排模型的排序能力展开。其中，双塔模型通过将用户侧与物品侧进行独立建模，并利用向量内积完成高效匹配，成为工业推荐系统中最经典的粗排模型之一；多塔模型则进一步通过引入额外的信息塔增强模型对于用户、物品以及交叉特征的建模能力；而蒸馏模型则通过知识迁移的方式，在不显著增加在线计算开销的情况下提升轻量级粗排模型的预测能力。

然而，无论是双塔模型还是多塔模型，其核心仍然依赖于用户 Embedding 与物品 Embedding 的独立表示学习。特别是在传统双塔结构中，用户侧与物品侧在网络较浅层便完成信息分离，最终仅通过向量内积完成匹配，这种结构虽然能够充分利用 Embedding 预计算实现极高的在线推理效率，但对于用户与物品之间复杂的高阶交互关系建模能力有限。在实际工业推荐场景中，用户兴趣往往受到历史行为、上下文环境、物品属性以及用户-物品之间长期交互关系等多种因素影响，仅依靠简单的向量匹配难以充分刻画这些复杂关系。

随着推荐系统业务规模不断扩大，用户兴趣日益多样化，候选物品数量也持续增长，工业界逐渐希望粗排阶段能够引入类似精排模型的深度特征交互能力，从而进一步提升排序精度。然而，传统精排 DNN 模型通常依赖大量用户-物品交叉特征，并需要进行复杂的非线性交互计算，如果直接应用于粗排阶段，需要对数千甚至上万个候选物品进行逐一打分，其在线计算成本难以满足工业级推荐系统对于低延迟、高吞吐的要求。

因此，近年来工业界开始探索更加高效的粗排 DNN 模型，希望在保持粗排阶段毫秒级在线推理效率的同时，引入更加丰富的特征交互能力。这类模型的核心思想是在传统双塔模型高效计算优势与精排 DNN 模型强表达能力之间进行权衡，通过不同形式的结构改进、特征选择以及模型协同优化，进一步缩小粗排模型与精排模型之间的效果差距。

本节主要介绍近年来工业界具有代表性的粗排 DNN 模型，包括双塔 DNN 模型、COLD 模型、FSCD 模型、IntTower 模型以及 RankTower 模型。这些模型虽然具体实现方式不同，但整体均围绕“**如何在粗排阶段引入更强**

的模型表达能力，同时满足工业级在线计算约束”这一核心问题展开。其中，双塔 DNN 模型在传统双塔结构基础上引入用户-物品交叉特征，通过轻量级 DNN 增强特征交互能力；COLD 模型进一步突破双塔模型必须依赖向量内积匹配的限制，使粗排阶段能够直接采用复杂 DNN 模型进行在线预测；FSCD 模型则从特征选择角度出发，通过可学习的特征筛选机制，在模型效果与计算成本之间实现更加灵活的平衡；IntTower 模型进一步探索双塔结构内部的特征交互机制，使双塔模型兼具效率与表达能力。

整体来看，工业界粗排 DNN 模型的发展过程，本质上体现了粗排系统从“高效向量匹配”逐渐向“高表达能力深度交互模型”演进的趋势。通过引入更加复杂的网络结构、更加丰富的特征交互以及更加先进的训练策略，现代粗排模型正在逐步逼近精排模型的预测能力，同时仍然保持工业级推荐系统对于实时性的严格要求。

11.4.1 双塔 DNN 模型

双塔 DNN 模型是在传统双塔模型基础上的一种增强方案，其核心思想如下：

定义 11.9

双塔 DNN 模型是在传统双塔结构基础上的改进方案。该模型在保留用户塔和物品塔独立表示学习以及 Embedding 预计算优势的基础上，在双塔顶层 Embedding 之外引入额外的用户-物品交叉特征，并将双塔输出的用户 Embedding、物品 Embedding 以及交叉特征 Embedding 共同输入轻量级 DNN 网络，最终输出模型预估的 pxtr 分数。通过这种方式，模型能够在保持粗排阶段在线推理效率的同时，进一步增强用户与物品之间高阶非线性交互关系的建模能力，从而提升排序精度。

传统双塔模型通常分别通过用户塔和物品塔学习用户 Embedding 与物品 Embedding，并通过向量内积计算用户与物品之间的匹配分数：

$$s(u,i) = \mathbf{e}_u^T \mathbf{e}_i \tag{11.70}$$

其中，$\mathbf{e}_u$ 表示用户 Embedding，$\mathbf{e}_i$ 表示物品 Embedding。由于用户塔和物品塔可以进行独立计算，因此物品侧 Embedding 能够提前离线预计算，并在在线阶段通过快速向量匹配完成大规模候选物品排序。然而，这种结构也限制了模型对于用户与物品之间复杂非线性交互关系的学习能力。

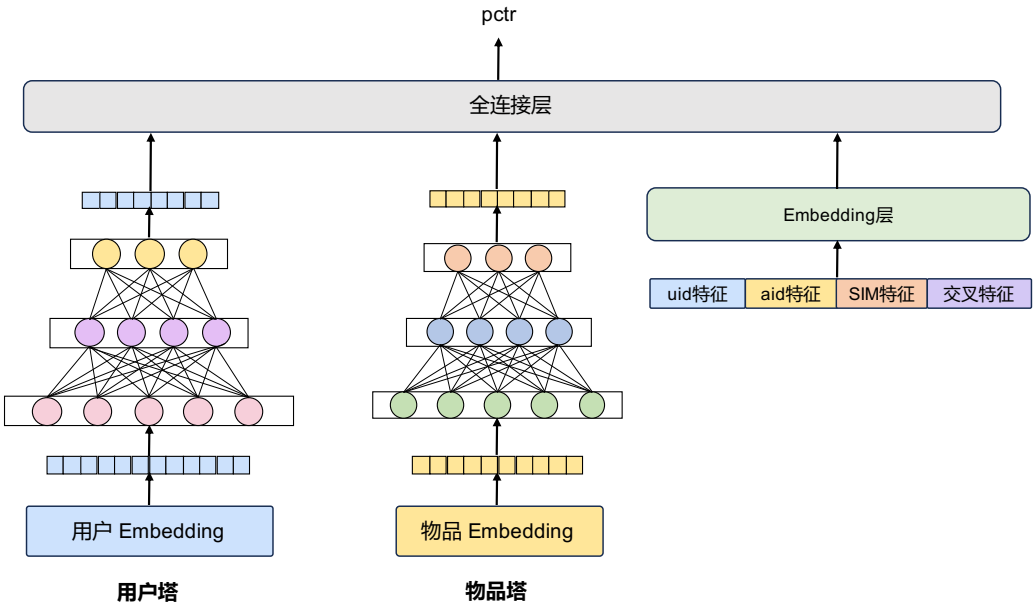

图 11.15: 双塔 DNN 模型结构示意图。

为了解决这一问题，工业界提出了双塔 DNN 模型，通过在双塔模型的顶层 Embedding 基础上进一步引入用户-物品交叉特征，并利用轻量级 DNN 网络完成最终预测。如图 11.15所示，双塔 DNN 模型仍然保留用户塔和

物品塔的独立表示学习过程，同时额外引入用户行为、物品属性以及用户-物品交叉信息作为补充输入。具体而言，假设用户侧输入特征经过用户塔得到表示：

$$\mathbf{h}_u = f_u(\mathbf{x}_u) \tag{11.71}$$

物品侧输入特征经过物品塔得到表示：

$$\mathbf{h}_i = f_i(\mathbf{x}_i) \tag{11.72}$$

此外，根据用户行为、物品属性以及用户-物品历史交互关系构造交叉特征表示：

$$\mathbf{h}_{cross} = f_c(\mathbf{x}_{u,i}) \tag{11.73}$$

随后，将用户塔输出、物品塔输出以及交叉特征表示进行融合：

$$\mathbf{h} = [\mathbf{h}_u, \mathbf{h}_i, \mathbf{h}_{cross}] \tag{11.74}$$

最终，通过多层全连接网络输出用户对物品的预测分数：

$$\hat{y} = f_{dnn}(\mathbf{h}) \tag{11.75}$$

相比传统双塔模型仅通过向量内积进行匹配，双塔 DNN 模型通过增加顶部 DNN 交互层，使模型能够学习用户 Embedding 与物品 Embedding 之间更加复杂的非线性关系，从而提升粗排阶段对于用户兴趣和物品相关性的建模能力。

需要注意的是，双塔 DNN 模型与前面章节介绍的多塔模型中的交叉特征塔存在一定区别。多塔模型通常将用户塔、物品塔以及交叉特征塔作为多个独立分支，并在顶层进行融合，因此交叉特征塔本身也参与完整的特征表示学习过程。而双塔 DNN 模型主要是在传统双塔结构的顶部增加轻量级交互网络，仅对最终 Embedding 进行融合，因此模型结构更加紧凑，在线推理成本更低，更适合粗排阶段大规模候选集排序场景。

在工业级推荐系统中，双塔 DNN 模型通常不会直接引入所有复杂交叉特征，而是根据计算成本和业务收益进行特征筛选。其中，一类常见做法是引入 SIM（Search-based Interest Model）检索得到的用户历史行为序列特征。由于用户历史行为序列通常包含大量长期兴趣信息，如果直接将完整行为序列输入粗排模型，会带来较大的计算开销。因此，工业系统通常首先通过 SIM 检索机制，从用户历史行为中筛选与当前候选物品最相关的若干行为，然后将这些检索结果作为用户-物品交叉特征输入双塔 DNN 模型。

例如，对于当前候选物品 i，SIM 模块可以从用户历史行为序列 $\mathcal{H}_u$ 中检索得到相关行为集合：

$$\mathcal{H}_u^i = SIM(\mathcal{H}_u, i) \tag{11.76}$$

随后，将检索得到的行为表示与当前物品 Embedding 进行交互，从而增强模型对于用户兴趣变化以及细粒度偏好的建模能力。除了 SIM 检索特征之外，工业系统中还会根据业务特点引入其他用户-物品交叉特征，例如用户历史点击类目与当前物品类目的匹配关系、用户历史行为与物品属性之间的组合特征等。这些特征能够进一步补充双塔模型中缺失的交互信息，使模型在粗排阶段获得更高的排序精度。

在工程部署方面，双塔 DNN 模型相比传统双塔模型增加了额外 DNN 计算，因此需要针对在线推理过程进行优化。例如，可以通过 Embedding 预计算减少重复计算，通过模型量化降低计算精度开销，通过 GPU 并行计算提升吞吐能力。此外，在大规模候选集场景下，还可以结合分布式计算、多线程并行以及批量推理等工程优化手段，使双塔 DNN 模型能够满足工业推荐系统对于低延迟、高并发的要求。

总体来看，双塔 DNN 模型是在传统双塔效率优势与精排 DNN 表达能力之间的一种折中方案。它没有完全放弃双塔结构带来的计算优势，而是在顶部增加有限的交互建模能力，因此成为工业界提升粗排模型效果的重要方向之一。

11.4.2 COLD 模型

COLD（Computing power cost-aware Online and Lightweight Deep pre-ranking）模型由阿里巴巴发表于 2020 年的论文《COLD: Towards the Next Generation of Pre-Ranking System》[11]。如图 11.16所示，与传统粗排模型主要采用双塔结构不同，COLD 提出粗排阶段同样可以采用复杂 DNN 模型进行预测，并通过模型设计与系统优化共同降低在线推理成本，从而实现模型效果与计算资源之间的灵活权衡。

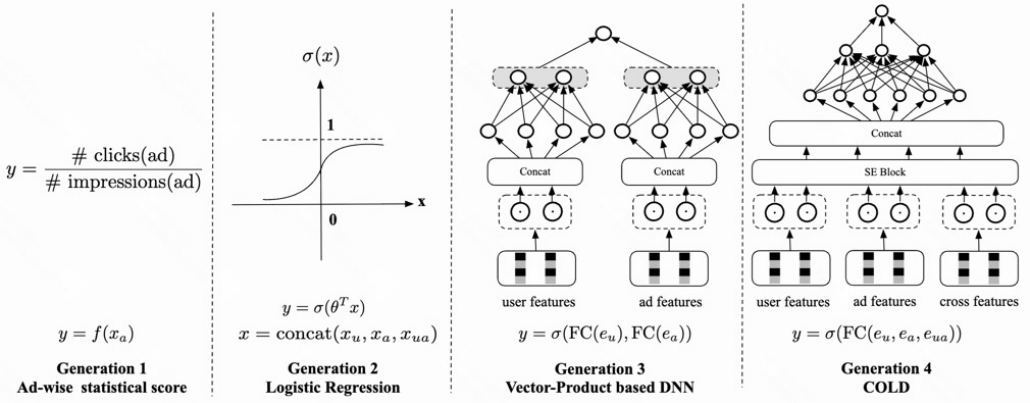

图 11.16: COLD 模型整体结构示意图。

COLD 模型的核心思想可以概括为两个方面。一方面，不再限制粗排模型必须采用向量内积（Inner Product）等轻量级网络结构，而是允许直接使用具有更强表达能力的深度神经网络，例如多层全连接网络（MLP）以及包含丰富交叉特征的排序模型，从而充分建模用户与物品之间复杂的高阶交互关系。另一方面，通过网络结构压缩以及工程优化等手段降低模型推理成本，使复杂 DNN 模型能够满足粗排阶段严格的在线时延要求。

从模型结构来看，COLD 采用与精排模型类似的全连接网络作为基础结构。用户特征、物品特征以及各种交叉特征首先分别完成 Embedding，然后拼接形成统一输入向量，最后输入多层全连接网络进行 CTR 预测，如图 11.16所示。相比传统双塔模型仅依赖用户 Embedding 与物品 Embedding 之间的内积进行匹配，COLD 能够直接学习任意形式的非线性交互，因此具有更强的模型表达能力。

然而，直接采用复杂 DNN 进行粗排预测会显著增加在线计算成本。为此，COLD 首先提出了一种基于 SE（Squeeze-and-Excitation）模块的特征组选择（Feature Group Selection）方法，用于自动筛选最重要的输入特征组。假设共有 M 个特征组，第 i 个特征组对应 Embedding 表示为 $\mathbf{e}_i$，则 SE 模块首先计算各特征组的重要性权重：

$$\mathbf{s} = \sigma\left(W[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_M] + \mathbf{b}\right), \tag{11.77}$$

其中，$\sigma(\cdot)$ 表示 Sigmoid 激活函数，W 和 $\mathbf{b}$ 均为可学习参数，$\mathbf{s}$ 表示各特征组的重要性评分。随后，每个特征组 Embedding 按照对应权重重新缩放：

$$\mathbf{v}_i = s_i \mathbf{e}_i \tag{11.78}$$

根据权重大小，系统进一步选择 Top-K 个最重要的特征组构建轻量级网络，并结合 GAUC、QPS 以及 RT（Response Time）等指标不断调整 K 的取值，从而在模型效果与计算资源之间寻找最优平衡。相比传统人工特征选择，这种方式能够更加灵活地控制模型复杂度，也是 COLD 能够兼顾效果与效率的重要原因之一。

除了网络结构优化之外，COLD 还结合大量工程优化进一步降低在线推理开销。例如，在系统层面，针对候选物品之间相互独立的特点，采用多级并行（Multi-level Parallelism）策略，将候选集合划分至多个线程甚至多个 GPU 并行计算；在特征计算阶段，采用列式计算（Column-based Computation）替代传统逐样本计算方式，从而提高 CPU 缓存命中率，并充分利用 SIMD 指令集加速特征处理；在神经网络推理阶段，则进一步采用 Float16 混合精度计算、GPU MPS（Multi-Process Service）等技术提升 GPU 吞吐能力。这些工程优化共同降低了复杂 DNN 模型的推理成本，使粗排阶段能够部署表达能力更强的深度网络。

除了模型结构之外，COLD 还构建了一套全在线（Fully Online）的粗排训练与服务架构。传统双塔粗排模型通常需要提前离线计算物品 Embedding 并构建索引，因此模型更新往往依赖离线流程，A/B 实验周期较长。而

COLD 模型采用在线训练与在线推理一体化架构，模型参数能够实时更新，不仅能够快速适应数据分布变化，也显著缩短了模型迭代与在线实验周期，更适合广告推荐、电商推荐等变化频繁的工业场景。

总体而言，COLD 模型最大的贡献并不在于提出了新的深度神经网络结构，而是在保持粗排阶段在线推理效率的前提下，通过特征组选择、工程优化以及全在线系统架构，使复杂 DNN 模型首次能够应用于工业级粗排系统。相比传统双塔模型，COLD 模型具有更强的特征交互能力和模型表达能力，同时又能够通过灵活控制模型复杂度满足不同业务场景对于时延与吞吐量的要求，因此成为近年来工业级粗排 DNN 模型的重要代表工作之一。在工业界落地和应用 COLD 模型架构的时候，落地难点通常在于线上推理时候的耗时增加，因此在落地的时候需要结合业务场景进行特征组选择以及工程优化，确保粗排阶段的在线推理延迟仍然满足毫秒级要求。一旦 COLD 模型成功落地，通常能够在不显著增加在线计算开销的前提下大幅提升传统基于双塔架构粗排模型的排序精度，从而进一步提升整体推荐系统的准确度和线上 A/B 实验收益。

11.4.3 FSCD 模型

FSCD（Feature Selection method based on feature Complexity and variational Dropout）模型源自阿里巴巴发表于 SIGIR 2021 的论文《Towards a Better Tradeoff between Effectiveness and Efficiency in Pre-Ranking: A Learnable Feature Selection based Approach》[7]。与前面介绍的 COLD 模型相比，FSCD 模型并没有重新设计粗排网络结构，而是将研究重点放在粗排模型的特征选择（Feature Selection）上，希望在保持交互式 DNN 模型表达能力的同时，自动筛选出最具价值且计算代价较低的特征，从而实现模型效果与在线推理效率之间更好的平衡。

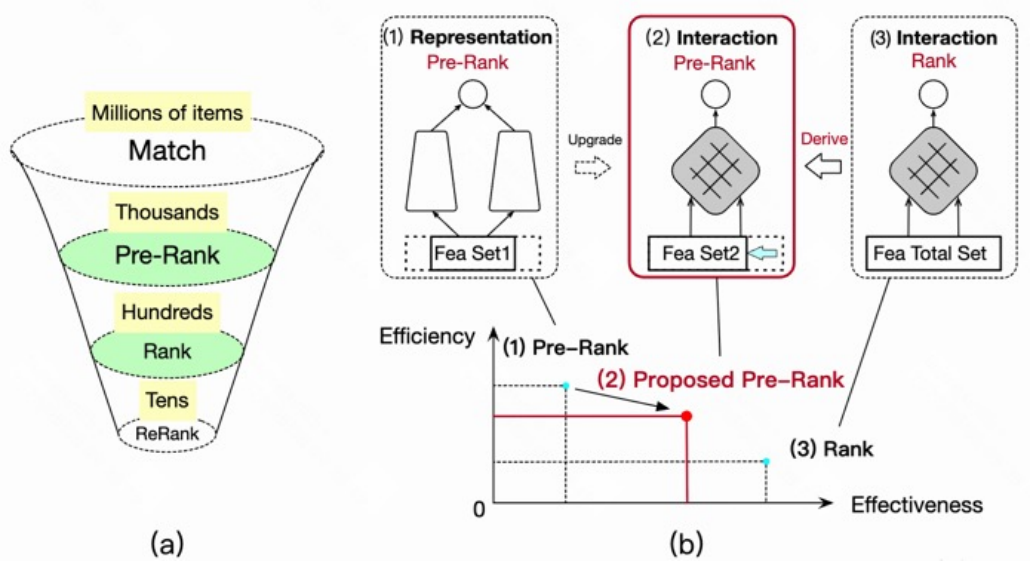

图 11.17: 多阶段推荐架构以及 FSCD 模型与传统双塔粗排、精排模型之间的关系。

如图 11.17所示，传统工业推荐系统通常采用召回、粗排、精排以及重排的多级架构。其中，精排模型拥有最丰富的特征和最强的模型表达能力，而粗排模型为了满足在线推理效率，通常仅使用少量特征或采用双塔结构，因此二者之间存在明显的能力差距。FSCD 认为，粗排模型并不一定需要采用完全不同于精排模型的网络结构，而是可以直接继承精排模型的交互式网络（Interaction-focused Architecture），仅通过减少输入特征集合来降低在线计算开销。这样不仅能够复用精排阶段已有的数据处理流程和模型结构，还能够缩小粗排与精排之间的优化目标差异，从而提升粗排模型的预测能力。

因此，FSCD 模型的整体结构如图 11.18所示。首先，用户特征、物品特征以及各种交叉特征分别完成 Embedding，然后进入可学习特征选择模块（Learnable Feature Selection）。该模块并不是人工指定保留哪些特征，而是在模型训练过程中自动学习每一个特征的重要程度，并结合不同特征对应的计算复杂度，最终筛选出最适合粗排阶段使用的特征集合。随后，仅保留筛选后的特征输入后续 DNN 网络完成 CTR 预测。

与传统特征选择方法最大的区别在于，FSCD 将特征选择过程直接融入模型训练之中。假设整个模型共有 M 个特征域（Feature Field），第 j 个特征域对应 Embedding 为 $\mathbf{v}_j$，FSCD 会为每一个特征域引入一个伯努利

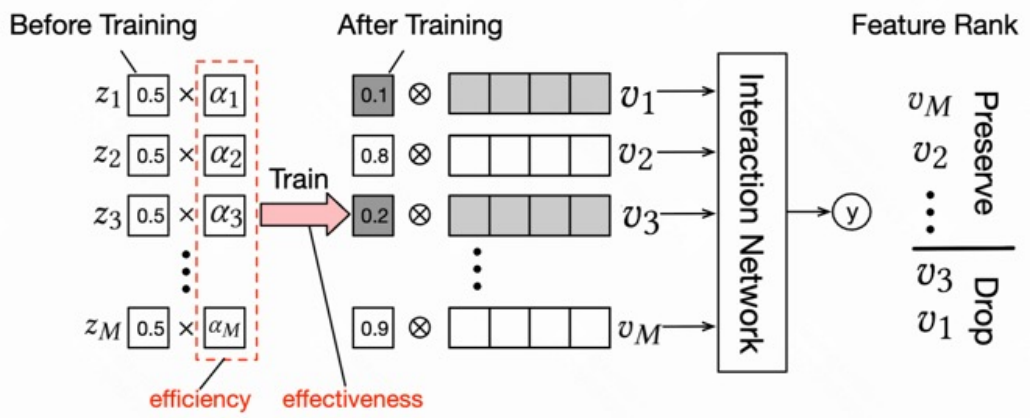

图 11.18: FSCD 模型整体结构示意图。

（Bernoulli）随机变量 z_j，用于表示该特征是否被保留：

$$z_j \sim \text{Bern}(\theta_j) \tag{11.79}$$

其中，$z_j = 1$ 表示保留该特征，$z_j = 0$ 表示舍弃该特征，θ_j 表示对应特征被保留的先验概率。为了同时考虑模型效果与计算效率，FSCD 进一步引入特征复杂度（Feature Complexity）的概念。对于第 j 个特征，其复杂度综合考虑在线计算开销、Embedding 维度以及特征空间大小等因素：

$$c_j = G(o_j, e_j, n_j) \tag{11.80}$$

其中，o_j 表示在线计算复杂度，e_j 表示 Embedding 维度，n_j 表示特征对应取值数量。实际工程中通常采用线性组合进行计算：

$$c_j = \gamma_1 o_j + \gamma_2 e_j + \gamma_3 n_j \tag{11.81}$$

其中，γ_1、γ_2 和 γ_3 用于控制不同复杂度因素的重要程度。随后，根据特征复杂度计算对应的先验保留概率：

$$\theta_j = 1 - \sigma(c_j) \tag{11.82}$$

其中，$\sigma(\cdot)$ 表示 Sigmoid 函数。由此可见，复杂度越高的特征，其保留概率越小，更倾向于在粗排阶段被移除；而计算成本较低的特征则更容易被保留下来。

在上述先验约束基础上，FSCD 模型进一步构建统一的训练目标，同时优化模型预测能力与特征选择策略，其整体优化目标为：

$$\mathcal{L} = -\frac{1}{N} \log P(\mathcal{D}|\mathbf{v}, \mathbf{w}, \mathbf{z}) + \frac{\lambda}{N} \left(\|\mathbf{v}\|_2^2 + \|\mathbf{w}\|_2^2 \right) + \frac{1}{N} \sum_{j=1}^{M} \alpha_j z_j \tag{11.83}$$

其中，第一项为 CTR 预测损失，第二项表示模型参数的 L_2 正则化，第三项则为特征选择正则项，用于控制最终保留特征数量。每个特征对应的正则化权重定义为：

$$\alpha_j = \log(1 - \theta_j) - \log(\theta_j) \tag{11.84}$$

由于 α_j 随着特征复杂度增加而不断增大，因此计算代价越高的特征将受到越强的正则化约束，更容易在训练过程中被自动舍弃。这也是 FSCD 模型区别于传统 Dropout 特征选择方法的重要创新，即首次将特征计算成本直接引入特征选择过程，而不仅仅关注预测效果。

由于 Bernoulli 随机变量不可导，无法直接利用梯度下降进行优化，因此论文进一步采用 Gumbel-Sigmoid 连续松弛（Continuous Relaxation）进行近似，将离散变量转换为连续可导变量：

$$z_j = \sigma \left(\frac{\log \delta_j - \log(1 - \delta_j) + \log u_j - \log(1 - u_j)}{t} \right) \tag{11.85}$$

其中，δ_j 表示可学习参数，$u_j \sim \text{Uniform}(0, 1)$ 表示随机噪声，t 为温度参数。训练过程中，δ_j 不断学习各特征的重要程度，当模型收敛后，按照 δ_j 的大小选择 Top-K 个特征，即可得到最终用于粗排阶段的特征集合。

完成特征筛选之后，FSCD 并不会直接部署训练得到的模型，而是进一步移除未被选中的特征，仅保留 Top-K

特征重新进行一次微调的过程，其优化目标退化为普通的 CTR 预测损失：

$$\mathcal{L} = -\sum_{i=1}^{N} \log p(y_i | f(x_i', \mathbf{v}', \mathbf{w}')) \tag{11.86}$$

其中，$\mathbf{v}'$ 与 $\mathbf{w}'$ 分别表示筛选后的 Embedding 参数与 DNN 参数，并利用第一阶段训练得到的参数作为初始化，从而进一步提升模型收敛速度与最终预测性能。

总的来说，FSCD 模型最大的创新在于提出了一种可学习的特征选择框架，将模型预测能力与在线计算成本统一纳入同一个优化目标，在一次训练过程中即可自动完成特征筛选与模型训练。相比传统依赖人工经验进行特征裁剪的方法，FSCD 能够更加准确地识别既有效又高效的特征集合；相比 COLD 主要关注网络结构压缩，FSCD 则从输入特征层面降低在线计算复杂度，两者具有较强的互补性。因此，FSCD 为后续工业界粗排模型中的自动特征选择（Automatic Feature Selection）提供了重要的研究思路，并成为工业推荐系统中兼顾效果与效率的代表性工作之一。

11.4.4 IntTower 模型

IntTower 模型源自于华为诺亚方舟实验室发表在 CIKM 2022 会议上的论文《IntTower: the Next Generation of Two-Tower Model for Pre-Ranking System》[6]。该模型针对传统双塔模型中用户塔与物品塔相互独立、仅在最终预测阶段进行交互的问题，在保留双塔模型高效推理优势的基础上，引入更加细粒度的用户与物品特征交互机制，从而进一步提升粗排模型的预测精度。

众所周知，传统双塔模型的核心优势在于用户塔与物品塔相互解耦，使得物品侧 Embedding 可以提前计算并进行离线存储，在线阶段只需要计算用户 Embedding，并通过向量相似度计算即可快速完成候选物品的打分匹配。然而，这种晚期交互（Late Interaction）的建模方式也限制了双塔模型本身的表达能力。由于用户塔与物品塔在各自独立的 DNN 网络中完成特征编码，直到最终计算向量内积或余弦相似度的阶段才发生交互，因此双塔模型难以充分利用用户特征与物品特征之间细粒度的交互信息。IntTower 模型对传统双塔模型进行了改进，其核心思想如下：

定义 11.10

IntTower 模型通过在双塔网络内部引入早期交互（*Early Interaction*），在尽可能保持用户与物品解耦结构和在线推理效率的前提下，增强双塔模型对于用户与物品交互关系的建模能力。

IntTower 模型的整体网络结构如图 11.19所示。给定用户侧特征和物品侧特征，首先分别经过 Embedding 层得到用户和物品的特征表示，并通过 Light-SE 模块对不同特征的重要性进行建模。在此基础上，用户特征和物品特征分别经过多层全连接网络进行编码。与传统双塔模型不同，IntTower 模型不会等到两个塔的最终 Embedding 才进行交互，而是从用户塔的多个中间层表示中提取特征，并与物品塔最后一层表示进行细粒度交互。最终，模型通过 FE-Block 得到用户与物品之间的相关性分数，同时利用 CIR 模块构造额外的对比学习目标，对用户与正样本物品之间的表示关系进行进一步约束。

11.4.4.1 Light-SE 模块

在工业级推荐系统中，不同特征对于用户行为预测的重要程度通常存在较大差异。例如，在电商推荐场景中，用户历史消费水平、用户近期行为等特征可能比性别等静态属性具有更强的预测能力；在直播推荐场景中，用户与主播的历史观看时长、历史打赏记录、是否关注等行为特征也可能比用户活跃度、主播登记等静态属性具有更强的预测能力。因此，如果将所有特征以同等权重输入进双塔网络中，模型可能根本无法捕捉与有效利用不同特征之间的重要性差异。针对这个问题，IntTower 模型首先通过 Light-SE（Lightweight SENET）模块来对不同特征进行重要性建模，从而获得更加精炼的用户侧和物品侧特征表示。

传统 SENET 通常需要通过两层全连接网络完成特征权重的学习，但对于粗排模型而言，额外的网络结构会增加在线推理延迟。因此，IntTower 模型对 SENET 这一模块进行了轻量化的设计。具体来说，对于包含 m 个

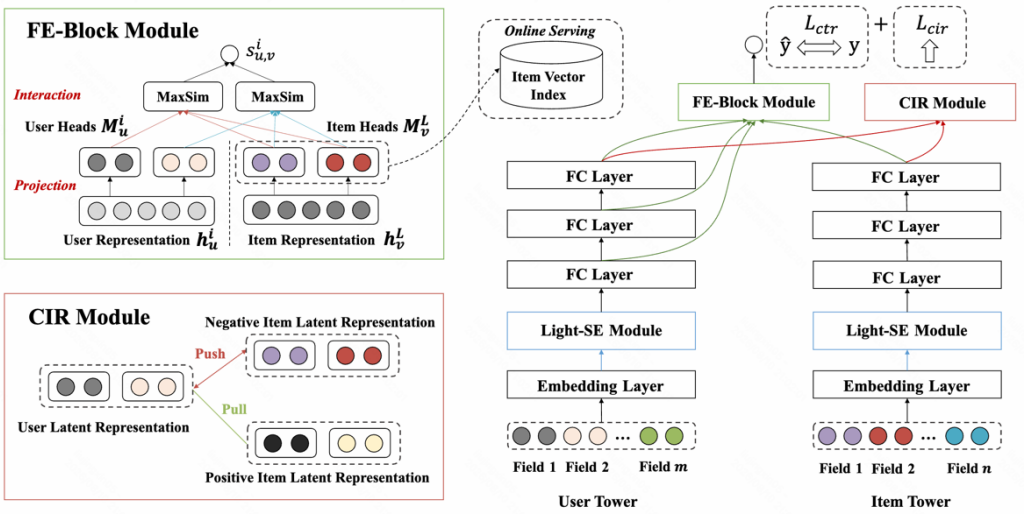

图 11.19: IntTower 模型整体结构示意图。

特征字段的用户侧或物品侧 Embedding，记第 i 个特征的 Embedding 为 $\mathbf{e}_i$，其维度为 d。Light-SE 模块首先通过 Mean Pooling 将每个特征 Embedding 压缩为一个标量统计量，即 $\mathbf{z} = [z_1, z_2, \ldots, z_m]$，其中第 i 个特征对应的统计量可以表示为

$$z_i = f_{sq}(\mathbf{e}_i) = \frac{1}{d}\sum_{t=1}^{d} e_{i,t} \tag{11.87}$$

随后，Light-SE 模块只使用一个全连接层计算不同特征的相对重要性，并采用 Softmax 函数代替传统 SENET 中的 Sigmoid 函数，使不同特征之间的权重具有归一化的相对关系。其特征权重可以表示为

$$\mathbf{k} = f_{ex}(\mathbf{z}) = \text{Softmax}(\mathbf{W}\mathbf{z} + \mathbf{b}) \tag{11.88}$$

其中，$\mathbf{W} \in \mathbb{R}^{m\times m}$ 和 $\mathbf{b} \in \mathbb{R}^m$ 分别表示可学习的变换矩阵和偏置向量，$\mathbf{k} = [k_1, k_2, \ldots, k_m]$ 表示不同特征字段的重要性权重。最后，将特征权重与原始 Embedding 进行逐特征相乘，得到经过特征重加权后的精炼特征表示：

$$\widetilde{\mathbf{e}} = f_{rw}(\mathbf{k}, \mathbf{e}) = [k_1\mathbf{e}_1, k_2\mathbf{e}_2, \ldots, k_m\mathbf{e}_m] \tag{11.89}$$

相比起完整的 SENET 结构，Light-SE 模块通过去除额外的全连接层，并采用更加轻量的特征统计和权重计算方式，在增加较少计算开销的情况下实现了对不同特征重要性的自适应建模。经过 Light-SE 模块处理之后，用户塔和物品塔能够获得更加精炼的特征表示，为后续的用户与物品之间细粒度的交互提供更加有效的输入。

11.4.4.2 FE-Block 模块

IntTower 通过细粒度早期特征交互（Fine-grained and Early Feature Interaction Block, FE-Block）模块来引入用户与物品特征之间的 *Early Interaction*，使用户和物品能够在网络的中间层进行更加细粒度的交互。这样可以有效缓解传统双塔模型中用户与物品特征交叉过晚的问题。

FE-Block 模块的一个重要设计原则如下：

笔记 FE-Block 模块在增强用户与物品交互的同时，尽可能保留双塔模型的用户与物品塔的解耦结构。具体而言，FE-Block 模块并不是直接将用户和物品的所有中间层表示拼接后输入一个复杂的交叉网络，而是将用户塔多个层级的表示分别映射到多个低维子空间中，再与物品塔最后一层表示进行细粒度的特征交互。由于物品侧只需要保存最终一层的多头表示，因此仍然可以对物品 Embedding 进行离线预计算和存储，从而降低在线推理成本。

假设用户塔包含 L 层全连接网络，第 i 层用户表示为 $\mathbf{h}_u^i$，通过 H 个不同的线性层将其映射到 H 个低维子空间，第 h 个子空间的表示为

$$\mathbf{m}_u^{i,h} = \mathbf{W}_u^{i,h}\mathbf{h}_u^i + \mathbf{b}_u^{i,h}, \quad h = 1, 2, \ldots, H \tag{11.90}$$

类似地，物品塔仅使用最后一层表示 $\mathbf{h}_v^L$ 进行投影，第 h 个子空间的物品表示为

$$\mathbf{m}_v^{L,h} = \mathbf{W}_v^{L,h}\mathbf{h}_v^L + \mathbf{b}_v^{L,h}, \quad h = 1, 2, \ldots, H \tag{11.91}$$

其中，$\mathbf{W}_u^{i,h}$、$\mathbf{b}_u^{i,h}$ 以及 $\mathbf{W}_v^{L,h}$、$\mathbf{b}_v^{L,h}$ 分别表示对应子空间的投影参数。通过多个 Head 分别表示不同的潜在语义空间，可以使模型从不同角度学习用户与物品之间的相关性。

在获得多头潜在表示之后，FE-Block 模块进一步计算用户第 i 层表示与物品最终层表示之间的细粒度交互。对于每一个用户侧 Head，FE-Block 模块从所有物品侧 Head 中选择相似度最高的一个进行匹配，并将所有用户侧 Head 对应的最大相似度进行求和，即

$$S_{u,v}^i = \sum_{h_u=1}^{H} \max_{h_v \in \{1,2,\ldots,H\}} \left\{ (\mathbf{m}_u^{i,h_u})^T \mathbf{m}_v^{L,h_v} \right\} \tag{11.92}$$

与直接对整个用户 Embedding 和物品 Embedding 进行一次内积相比，这种方式能够在更细粒度的表示空间中捕捉用户与物品之间的局部匹配关系。

进一步地，FE-Block 模块将用户塔多个中间层产生的交互分数进行聚合，得到最终的用户与物品之间的相关性预测分数

$$\hat{y} = \sum_{i=1}^{L} S_{u,v}^i \tag{11.93}$$

因此，FE-Block 模块实际上同时利用了用户塔中不同层级的表示。较浅层的表示能够捕捉相对底层的特征交互信息，而较深层的表示则能够提供更加抽象的用户兴趣与物品语义信息。通过多层表示的联合交互，IntTower 模型能够获得更加丰富的用户与物品匹配信号。

值得注意的是，FE-Block 模块中的核心交互操作本身不包含额外的可学习参数，主要依赖向量相似度计算和最大化相似度操作完成交互。这种设计对于粗排场景尤为重要，因为复杂的交叉网络虽然能够提升模型表达能力，但通常会显著增加在线计算成本。与此同时，物品塔仅使用最后一层表示参与交互，因此 $\mathbf{m}_v^L$ 可以提前离线计算并进行存储，在线阶段只需要计算用户侧表示，并与候选物品的预计算表示完成细粒度匹配，从而在增强交互能力的同时保持较高的推理效率。

从整体上看，FE-Block 模块将传统双塔模型的 *Late Interaction* 转变为更加细粒度的 *Early Interaction*，但并没有完全破坏双塔模型的用户与物品解耦结构，这也是 IntTower 模型能够应用于工业级粗排场景的一个重要原因。

11.4.4.3 CIR 模块

FE-Block 模块主要从显式角度增强用户与物品之间的特征交互，但仅依赖监督标签进行显式交互建模，仍然可能无法充分约束用户表示与正样本物品表示之间的潜在关系。为此，IntTower 模型进一步引入对比交互正则化（Contrastive Interaction Regularization, CIR）的模块，通过对比学习构造额外的自监督交互信号，从隐式角度增强用户与物品之间的信息关联。

在 FE-Block 模块中，用户塔和物品塔最后一层的多头潜在表示分别记为 $\mathbf{M}_u^L$ 和 $\mathbf{M}_v^L$。对于训练样本中的用户与正样本物品对 (u, v)，IntTower 模型将其视为正样本对，而将同一用户对应的其他物品表示视为负样本，从而构造用户与物品之间的对比学习任务。首先定义用户与物品最终潜在表示之间的余弦相似度为：

$$\text{sim}(\mathbf{M}_u^L, \mathbf{M}_v^L) = \frac{(\mathbf{M}_u^L)^T \mathbf{M}_v^L}{\|\mathbf{M}_u^L\| \|\mathbf{M}_v^L\|} \tag{11.94}$$

在此基础上，采用 InfoNCE 构造对比交互正则化损失，使用户表示与正样本物品表示之间的相似度尽可能增大，同时降低与负样本物品表示之间的相似度。其损失函数可以表示为

$$L_{\text{cir}} = -\frac{1}{Q}\sum_{(u,v)\in Q}\log\frac{\exp(\text{sim}(\mathbf{M}_u^L,\mathbf{M}_v^L)/\tau)}{\sum_{(u',v')\in N}\exp(\text{sim}(\mathbf{M}_{u'}^L,\mathbf{M}_{v'}^L)/\tau)} \tag{11.95}$$

其中，Q 表示正样本数量，N 表示参与对比学习的样本集合，τ 为温度系数。通过引入 CIR，模型不再完全依赖点击、购买等显式行为标签，而是进一步利用用户与正样本物品之间的表示一致性构造辅助监督信号，从而使用户塔和物品塔的潜在表示在训练过程中形成更加紧密的关联。

需要注意的是，CIR 模块并不是替代 FE-Block 模块，而是作为其补充。FE-Block 模块通过最大化相似度显式建模用户与物品之间的细粒度交互，而 CIR 模块则通过对比学习在表示空间中进一步约束用户与正样本物品的潜在关系。因此，IntTower 模型分别从**显式交互**和**隐式交互**这两个角度增强了传统双塔模型的表达能力。

11.4.4.4 IntTower 模型训练

IntTower 模型最终采用监督学习与对比学习相结合的方式进行联合训练。对于每一个训练样本，首先利用模型预测分数 $\hat{y}$ 与真实标签 y 计算 Log Loss，以学习用户与物品之间的行为预测关系，其监督学习损失可以表示为：

$$L_{\text{ctr}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{11.96}$$

在此基础上，将 CIR 模块产生的对比交互正则化损失加入整体训练目标，并增加模型参数的 L_2 正则项，最终得到 IntTower 模型的联合优化目标为：

$$L = L_{\text{ctr}} + \lambda_1 L_{\text{cir}} + \lambda_2\|\Theta\|_2^2 \tag{11.97}$$

其中，λ_1 用于控制对比交互正则化的影响，λ_2 用于控制模型参数正则化强度，Θ 表示 IntTower 模型的全部可学习参数。通过联合优化监督预测损失与对比学习损失，IntTower 模型既能够直接学习用户行为标签所提供的排序信息，又能够利用额外的表示空间约束增强用户与物品之间的潜在交互关系。

总体来看，IntTower 模型并没有简单地通过增加一个复杂的交叉网络来换取模型效果，而是针对粗排阶段对计算效率的严格要求，对用户与物品的交互机制进行了专门设计。其中，Light-SE 模块以较低的计算成本筛选重要特征，FE-Block 模块通过无额外参数的最大化相似度机制实现细粒度的早期交互，CIR 模块则利用对比学习进一步增强隐式交互。三者共同构成了 IntTower 模型的核心架构，使其在保留双塔模型较高在线推理效率的同时，显著缓解传统双塔模型中用户与物品交互不足的问题。

11.5 粗排模型多兴趣建模

与召回模型类似，工业级推荐系统中的粗排模型同样存在多兴趣建模的需求。对于一个真实用户而言，其兴趣通常并不是单一且固定的，而是可能同时覆盖多个不同的内容主题、消费场景以及行为目标。例如，一个用户可能既对体育内容感兴趣，也会关注美食、旅游等内容；同时，其点击、收藏、购买等不同层次的行为背后，也可能对应着不同的兴趣偏好。如果仍然使用单一的用户 Embedding 对这些复杂兴趣进行统一表示，不同兴趣之间的信息可能会相互混合甚至相互干扰，导致某些相对弱但具有实际价值的兴趣被主兴趣淹没，进而影响用户与候选物品之间匹配分数的准确性。

因此，在粗排模型中引入多兴趣建模，可以针对用户的不同兴趣维度学习更加细粒度的表示，并在不同候选物品或不同预测目标下，动态选择与当前场景更加相关的兴趣信息。相比使用一个固定用户 Embedding 与所有候选物品进行匹配，多兴趣表示能够更加准确地刻画用户在不同场景下的行为偏好，从而增强粗排模型对于复杂用户兴趣的建模能力，提升候选物品筛选阶段的排序准确性。

从工业界推荐算法优化的角度来看，粗排模型中的多兴趣建模并不仅仅是为了增加模型结构的复杂度，其更重要的价值在于贴合具体业务场景的特点，为粗排模型引入新的增量信息。例如，可以围绕用户的不同内容

兴趣、不同商品类别偏好、不同消费意图，或者点击、转化等不同层次的行为目标，对用户表示进行更加细粒度的拆分和建模。由于这种建模方式能够更加准确地刻画用户兴趣，因此通常能够直接提升粗排模型的预测能力。

进一步从线上 AB 实验的角度来看，多兴趣建模往往也具有较强的实际业务价值。相比单纯扩大 DNN 网络规模或者增加模型参数，多兴趣建模更多是从用户行为与业务场景中挖掘新的信息增量。当新的兴趣表示能够有效补充原有单一用户 Embedding 难以表达的信息时，通常能够在点击、时长、转化等核心业务指标上获得较为明显的线上收益。因此，在工业级推荐系统中，多兴趣建模可以理解为一种兼具模型表达能力提升与业务信息增量挖掘价值的重要优化方向。

本节主要介绍两个具有代表性的粗排多兴趣建模方法，分别是腾讯在 KDD 2022 会议论文中提出的 MVKE 模型，以及腾讯在 CIKM 2025 会议论文中提出的 HIT 模型。其中，MVKE 模型主要通过多个虚拟核专家对用户的不同兴趣维度进行建模，并利用候选目标相关的门控机制动态聚合不同兴趣表示；HIT 模型则进一步针对更加复杂的用户兴趣建模场景进行优化。下面首先介绍 MVKE 模型。

11.5.1 MVKE 模型

MVKE 模型源自于腾讯在 KDD 2022 会议上发表的论文《Mixture of Virtual-Kernel Experts for Multi-Objective User Profile Modeling》[13]。MVKE 的全称为 Mixture of Virtual-Kernel Experts，即虚拟核专家混合模型，其核心目标是解决传统双塔模型使用单一用户 Embedding 表示多样化用户兴趣时存在的表达能力不足问题。

传统双塔模型通常将用户的所有特征输入用户塔，并最终输出一个固定的用户 Embedding。该 Embedding 需要同时压缩用户在不同内容主题、不同兴趣方向以及不同行为目标下的偏好信息，因此容易造成不同兴趣之间的相互干扰。与此同时，传统双塔模型中的用户表示通常独立于目标物品或目标标签生成，用户塔与物品塔之间缺少充分的信息交互，也限制了模型针对不同目标动态选择用户兴趣的能力。

针对上述问题，MVKE 模型仍然保持传统双塔模型的整体高效架构，但在用户塔中引入多个 Virtual-Kernel Experts（VKE），即虚拟核专家；并进一步通过 Virtual-Kernel Gate（VKG），即虚拟核门控机制，将不同专家学习到的用户兴趣进行动态聚合。MVKE 模型的核心思想可以概括为：

定义 11.11

MVKE 模型通过多个虚拟核专家分别建模用户不同维度的隐式兴趣，并使用目标物品或目标标签相关的虚拟核门控机制，对多个专家输出进行动态加权组合，从而生成与当前目标更加匹配的用户表示。虚拟核同时参与用户侧兴趣建模与目标相关的兴趣聚合过程，因此能够作为连接用户塔与目标塔之间的信息桥梁，在增强多兴趣建模能力的同时保持双塔模型较高的在线推理效率。

MVKE 模型的整体结构如图 11.20所示。从整体架构来看，MVKE 模型仍然采用类似双塔模型的结构，其中目标塔的网络结构与传统双塔模型基本保持一致，而用户塔则由多个 VKE 组成。所有 VKE 共享底层用户特征输入，但每个 VKE 内部维护一个不同的可学习虚拟核，并在虚拟核的引导下从相同的用户特征中提取不同维度的兴趣信息。

具体来说，假设用户特征经过 Embedding 层后得到特征表示集合 $\mathbf{E}_u = [\mathbf{e}_{u1}, \mathbf{e}_{u2}, \ldots, \mathbf{e}_{um}]$。对于第 k 个 VKE，模型维护一个可学习的虚拟核 $\mathbf{W}_{VK}^k$。该虚拟核可以被视为一个隐式的兴趣查询，用于从用户的多个特征表示中选择与当前兴趣维度更加相关的信息。

在 VKE 内部，虚拟核首先经过非线性映射得到注意力机制中的 Query，而用户特征 Embedding 分别映射为 Key 和 Value，即

$$\mathbf{Q} = \sigma(\mathbf{W}_Q^T \mathbf{W}_{VK}^k + \mathbf{b}_Q), \quad \mathbf{K} = \sigma(\mathbf{W}_K^T \mathbf{E}_{uf_i} + \mathbf{b}_K), \quad \mathbf{V} = \sigma(\mathbf{W}_V^T \mathbf{E}_{uf_i} + \mathbf{b}_V) \tag{11.98}$$

其中，$\sigma(\cdot)$ 表示非线性激活函数，$\mathbf{W}_Q$、$\mathbf{W}_K$ 和 $\mathbf{W}_V$ 表示不同的投影矩阵，$\mathbf{b}_Q$、$\mathbf{b}_K$ 和 $\mathbf{b}_V$ 表示对应的偏置项。随后，第 k 个 VKE 利用注意力机制，根据虚拟核与不同用户特征之间的相关性，对用户特征进行加权聚合：

$$\mathbf{C}_{VKE}^k = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}}\right)\mathbf{V} \tag{11.99}$$

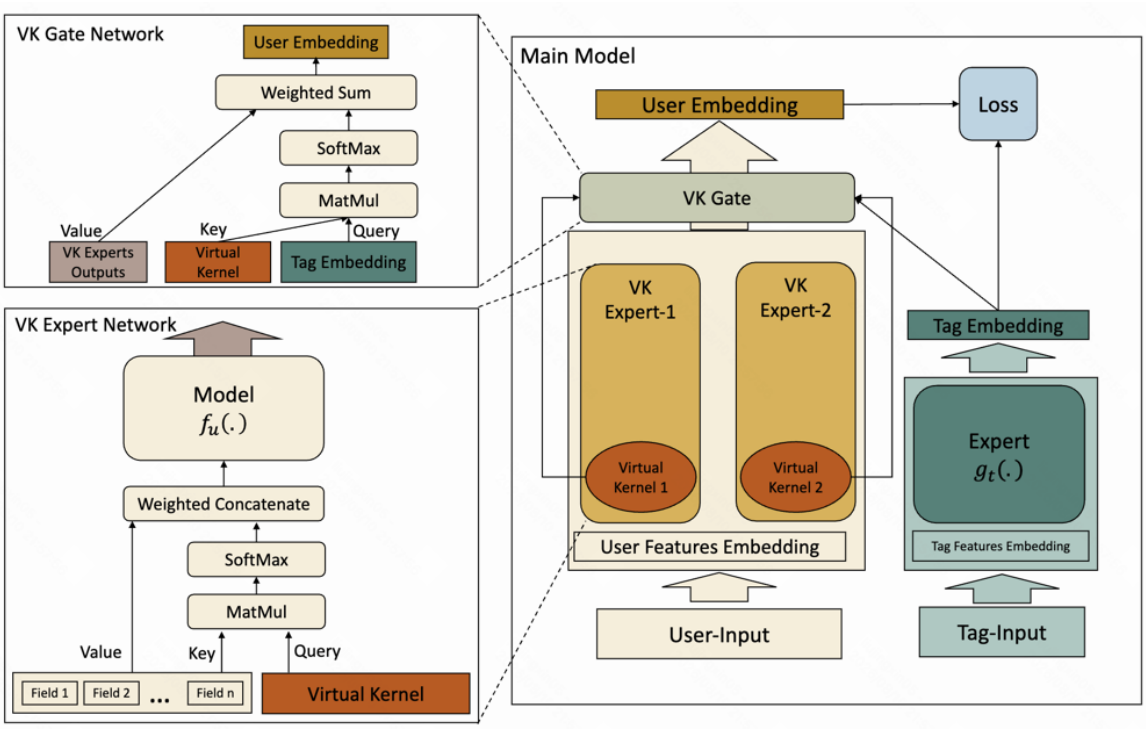

图 11.20: MVKE 模型整体结构示意图。

与直接将所有用户特征简单拼接后输入 DNN 不同，VKE 通过不同的虚拟核对用户特征进行不同方式的聚合，因此不同 VKE 可以分别关注用户兴趣的不同侧面。然后，每个 VKE 进一步通过一个特征建模网络 $f_u^k(\cdot)$ 生成对应的用户兴趣表示：

$$\mathbf{E}_{ui}^k = f_u^k(\mathbf{C}_{VKE}^k) \tag{11.100}$$

其中，$f_u^k(\cdot)$ 可以采用简单的加权聚合网络，也可以根据实际业务需求叠加 DeepFM、xDeepFM 等更复杂的特征交互结构。最终，第 k 个 VKE 输出一个对应于某一隐式兴趣空间的用户 Embedding $\mathbf{E}_{ui}^k$。

需要注意的是，单个 VKE 学习到的兴趣表示本身是隐式的，并不直接对应某一个具有明确语义的真实兴趣类别。因此，MVKE 进一步引入 Virtual-Kernel Gate（VKG）模块，根据当前目标物品或目标标签动态选择不同 VKE 输出的重要性。

具体来说，VKG 以目标物品的 Embedding $\mathbf{E}_{T_i}$ 作为 Query，以多个虚拟核 $\{\mathbf{W}_{VK}^k\}$ 作为 Key，以多个 VKE 输出 $\{\mathbf{E}_{ui}^k\}$ 作为 Value，通过注意力机制计算不同兴趣表示对于当前目标物品的重要程度，并得到最终与目标物品相关的用户表征，即：

$$\mathbf{E}_{ui} = \sum_k \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{E}_{T_i})\mathbf{K}(\mathbf{W}_{VK}^k)^T}{\sqrt{d_K}}\right)\mathbf{V}(\mathbf{E}_{ui}^k) \tag{11.101}$$

通过这一机制，同一个用户在面对不同目标物品或标签时，可以动态组合不同的兴趣表征。例如，当目标物品属于体育类别时，模型可能更加关注用户对应体育兴趣的 VKE 输出；而当目标物品属于美食类别时，则可以提升其他兴趣专家的权重。因此，MVKE 模型最终生成的用户表征不再是传统双塔模型中的固定单一 Embedding，而是能够根据目标进行动态调整的多兴趣表征。

从模型结构的角度来看，虚拟核是 MVKE 模型最核心的设计。它一方面用于指导不同 VKE 从用户特征中提取不同维度的兴趣信息，另一方面又作为 VKG 进行目标相关兴趣选择时的重要桥梁。因此，虚拟核实际上连接了用户侧的多兴趣建模与目标侧的信息需求，在一定程度上缓解了传统双塔模型中用户塔与目标塔完全独立的问题。

需要注意的是，MVKE 模型不仅可以应用于单任务场景，还可以进一步扩展到多任务推荐场景。在多任务

设置下，MVKE 模型可以为不同任务配置不同数量和不同组合的 VKE。其中一部分 VKE 可以作为多个任务共享的专家，用于学习不同任务之间具有共性的用户兴趣；另一部分 VKE 则可以作为任务特定专家，仅服务于某一个任务，从而保证不同任务之间的建模差异。在多任务场景下，MVKE 模型的整体结构如图 11.21所示。

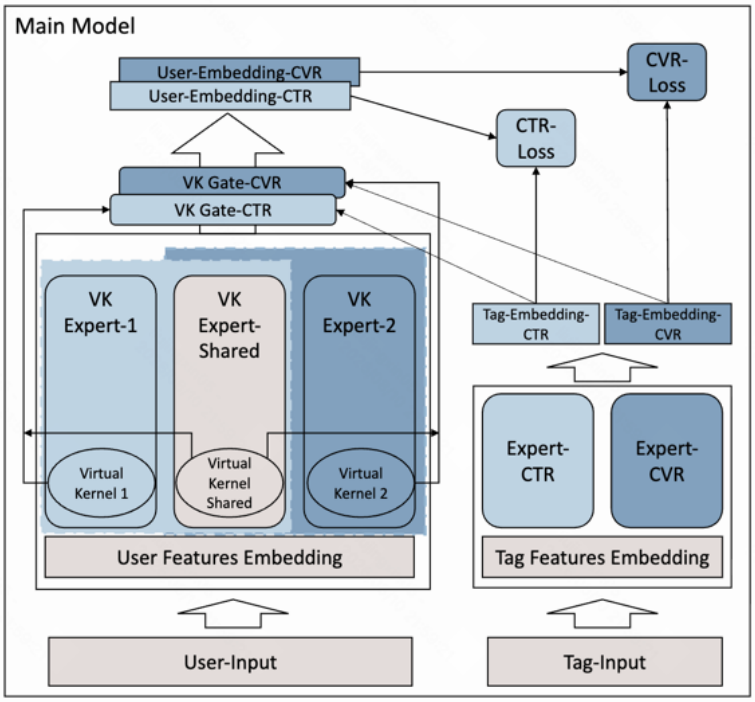

图 11.21: 多任务 MVKE 模型整体结构示意图。

例如，在点击与转化这两个具有明显行为序列关系的任务中，点击任务通常可以使用一组基础兴趣专家，而更深层的转化任务则可以在这些基础兴趣专家的基础上使用更多专家，从而同时利用浅层行为信息与更加复杂的用户意图信息。假设共享 VKE 集合为 $\mathcal{K}_s$，点击任务和转化任务的特定专家集合分别为 $\mathcal{K}_{ctr}$ 和 $\mathcal{K}_{cvr}$，则多任务训练的整体损失可以表示为：

$$L_{MTL} = L_{ctr} + L_{cvr} \tag{11.102}$$

其中，每个任务分别使用共享专家与对应任务特定专家生成用户表示，并根据具体预测目标计算相应的任务损失。这种设计使 MVKE 能够同时兼顾专家的共享能力与任务差异化建模能力。

在线推理效率方面，MVKE 模型仍然继承了双塔模型的核心优势。传统双塔模型可以分别预计算用户 Embedding 与目标物品的 Embedding，再通过向量内积完成快速匹配，因此避免了对所有用户与目标物品组合重复执行复杂的 DNN 推理。MVKE 模型虽然引入了多个 VKE，但不同 VKE 的用户兴趣表示可以提前计算并存储，在线阶段只需要结合目标物品的 Embedding 计算 VKG 的注意力权重，再对多个 VKE 输出进行快速聚合。

因此，MVKE 模型相比传统双塔模型主要增加了用户侧多组兴趣表示的存储开销，而不会显著改变整体的在线计算复杂度。假设 VKE 数量为 k，则模型需要为每个用户存储 k 组兴趣表示，其空间复杂度会相应增加，但由于用户侧 VKE 输出可以离线或提前计算，在线阶段仍然能够保持较高的推理效率。这种以一定存储资源换取更强用户兴趣表达能力的设计，也较符合工业级推荐系统中的实际工程需求。

整体来看，MVKE 模型可以被视为对传统双塔模型单一用户表示的一种重要扩展。传统双塔模型通常将用户复杂且多样化的兴趣压缩为一个固定 Embedding，而 MVKE 模型通过多个虚拟核专家（VKE）分别建模不同兴趣维度，再利用目标相关的虚拟核门控（VKG）机制动态选择不同兴趣表示，使用户 Embedding 能够随着目标发生变化。在保持双塔模型高效推理能力的同时，MVKE 模型进一步增强了用户多兴趣建模能力以及用户与目标之间的信息交互能力，因此特别适合用户兴趣多样、业务目标复杂的工业级粗排场景。

11.5.2 HIT 模型

HIT 模型是腾讯在 CIKM 2025 会议上发表的论文《HIT Model: A Hierarchical Interaction-Enhanced Two-Tower Model for Pre-Ranking Systems》[14] 中提出的一种面向工业级粗排系统的双塔增强模型。HIT 模型的全称为 Hierarchical Interaction-Enhanced Two-Tower，即分层交互增强的双塔模型，其核心目标是在保持传统双塔模型高效推理能力的基础上，进一步增强用户与物品之间的交互建模能力，从而提升粗排阶段对于复杂用户兴趣以及物品多维属性的刻画能力。

传统双塔模型虽然通过用户塔与物品塔的独立编码方式，实现了 Embedding 预计算以及高效向量匹配，但由于用户侧与物品侧信息在编码过程中完全解耦，模型难以充分捕获二者之间复杂的语义关联关系。同时，传统双塔模型通常采用单一 Embedding 表示用户兴趣以及物品属性，这种方式难以适应工业推荐系统中用户兴趣多样化以及物品属性多维化的实际特点。因此，如何在不破坏双塔模型在线效率优势的情况下，引入更加丰富的交互信息成为粗排模型优化的重要方向。

HIT 模型针对上述问题提出了一种分层交互增强机制，其核心思想如下：

定义 11.12

HIT 模型在传统双塔结构的基础上，引入粗粒度生成（Coarse-grained Generation）模块以及细粒度匹配（Fine-grained Matching）模块。其中，粗粒度生成模块通过生成器学习用户与物品之间的高层语义关联，并将对侧塔的信息融入当前塔表示中；细粒度匹配模块通过多头表示器（Multi-head Representer）将用户兴趣和物品属性映射到多个潜在子空间，通过多兴趣与多属性之间的细粒度匹配提升模型表达能力。在保持物品 Embedding 可预计算的前提下，HIT 模型进一步增强了双塔模型对于复杂用户-物品交互关系的建模能力。

HIT 模型整体结构如图 11.22所示。相比传统双塔模型仅通过用户 Embedding 与物品 Embedding 之间的向量相似度进行匹配，HIT 模型通过引入分层交互机制，使模型能够同时学习用户与物品之间的粗粒度语义关联以及细粒度兴趣匹配关系。具体来说，粗粒度生成模块主要负责建立用户与物品之间的隐式关联，使用户塔能够感知物品侧信息，物品塔能够感知用户侧信息；而细粒度匹配模块则进一步将用户和物品表示投影到多个不同的潜在空间，从多个兴趣维度计算用户与物品之间的匹配程度。

从图 11.22可以明显地看出，HIT 模型在整体网络结构的设计上融合了近年来双塔模型增强交互能力的多种优化思想。其中，在跨塔信息融合方面，HIT 模型受到 DAT 模型的启发，通过在用户塔和物品塔中分别引入对侧塔的高层语义表示，使用户表示和物品表示不再完全独立，从而在保持双塔结构以及 Embedding 预计算优势的同时，增强用户与物品之间的信息交互能力。

除此之外，HIT 模型进一步融合了 IntTower 和 MVKE 模型中的多头表征思想，在用户塔和物品塔中分别生成多个不同 Head 的潜在空间表示，使模型能够从多个兴趣维度刻画用户偏好，同时从多个属性维度描述物品特征。在匹配阶段，HIT 模型采用类似 IntTower 中的 Max Similarity 机制，在不同用户兴趣子空间与物品属性子空间之间寻找最佳匹配关系，并聚合多个维度上的匹配结果，从而进一步增强用户与物品之间细粒度交互关系的建模能力。整体来看，HIT 模型通过跨塔语义融合与多兴趣细粒度匹配相结合的方式，在不牺牲双塔模型在线推理效率的前提下，有效提升了粗排模型对于复杂用户兴趣和物品属性的表达能力。

11.5.2.1 粗粒度生成模块

传统双塔模型最大的限制在于用户塔与物品塔完全独立，使得用户表示仅包含用户自身信息，物品表示仅包含物品自身信息，两侧之间缺少有效的信息交互。为了缓解这一问题，HIT 模型提出了粗粒度生成模块，通过 Generator 网络提前生成包含对侧信息的高层语义表示，从而在保持双塔结构的同时建立用户与物品之间的隐式联系。

具体而言，对于用户塔而言，Generator 以用户静态特征作为输入，生成模拟物品侧信息的表示；对于物品塔而言，Generator 则生成模拟用户侧信息的表示。由于生成过程主要用于学习用户与物品之间稳定的高层语义

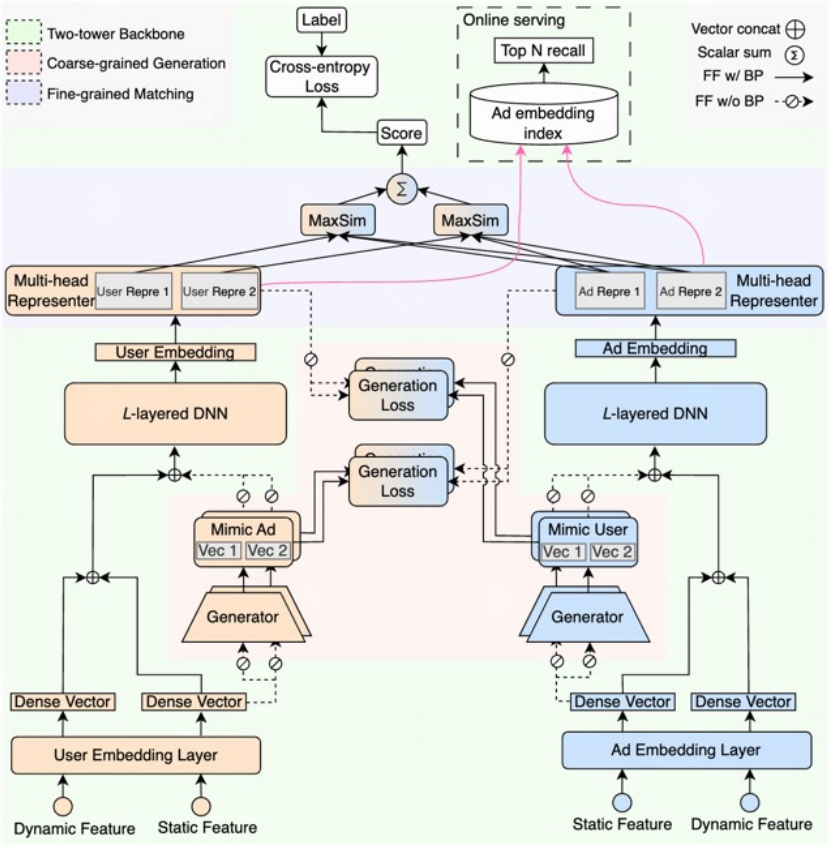

图 11.22: HIT 模型整体结构示意图。

关联，因此 HIT 模型仅采用静态特征作为 Generator 输入，从而避免引入实时行为特征导致线上计算开销增加。假设 Generator 数量为 K，第 k 个 Generator 输出的模拟表示可以表示为：

$$\mathbf{r}_m^k = g_k(\mathbf{x}_b^\delta), \quad k = 1, 2, \cdots, K \tag{11.103}$$

其中，δ 表示用户侧或物品侧，$\mathbf{x}_b^\delta$ 表示对应侧的静态特征输入，$\mathbf{r}_m^k$ 表示第 k 个 Generator 生成的对侧语义表示。在广告推荐场景中，由于训练样本通常包含正负反馈，因此 HIT 模型分别设计正样本 Generator 和负样本 Generator，用于学习目标物品与非目标物品之间的差异化表示。

在得到 Generator 生成的表示后，HIT 模型首先对其进行 L2 归一化，然后与原始用户或物品 Embedding 进行拼接，形成增强后的输入表示：

$$\mathbf{e}' = f(\mathbf{r}_m^1, \mathbf{r}_m^2, \cdots, \mathbf{r}_m^K, \mathbf{e}) \tag{11.104}$$

随后，将增强后的表示输入 DNN 网络得到最终用户表示 $\mathbf{h}_u$ 以及物品表示 $\mathbf{h}_a$。通过这种方式，用户塔和物品塔虽然仍然保持结构独立，但其表示中已经包含部分来自另一侧的信息，从而增强了双塔模型对于用户-物品关系的感知能力。

11.5.2.2 细粒度匹配模块

在粗粒度生成模块的基础上，HIT 模型进一步提出细粒度匹配模块，用于捕获用户多兴趣以及物品多属性之间更加精细的匹配关系。该模块的核心思想与多兴趣建模类似，即认为单一用户 Embedding 难以完整表示用户复杂多样的兴趣，因此需要将用户表示映射到多个潜在子空间，每个子空间负责刻画用户的一种潜在兴趣，同时物品侧也通过多个子空间表示不同属性维度。

具体来说，HIT 模型通过 Multi-head Representer 将用户表示投影到 J 个不同的潜在空间，第 j 个用户兴趣表示可以表示为：

$$\mathbf{r}_u^{(j)} = \mathbf{W}_u^{(j)} \mathbf{h}_u + \mathbf{b}_u^{(j)}, \quad j = 1, 2, \cdots, J \tag{11.105}$$

同理，对于物品侧表示：

$$\mathbf{r}_a^{(j)} = \mathbf{W}_a^{(j)}\mathbf{h}_a + \mathbf{b}_a^{(j)}, \quad j = 1, 2, \cdots, J \tag{11.106}$$

其中，$\mathbf{W}_u^{(j)}$ 和 $\mathbf{W}_a^{(j)}$ 分别表示用户侧和物品侧第 j 个子空间的映射矩阵。通过这种多头表示机制，HIT 模型能够从多个兴趣维度理解用户，同时从多个属性维度刻画物品。

在计算用户与物品匹配分数时，HIT 模型并不是简单计算两个整体 Embedding 之间的余弦相似度，而是在不同兴趣子空间之间寻找最佳匹配关系。最终预测分数的定义仍然使用了 IntTower 模型中基于 Max Similarity 的计算方式，即：

$$\hat{y} = \sum_{j_u=1}^{J} \max_{j_a \in \{1,2,\cdots,J\}} \{(\mathbf{r}_u^{(j_u)})^T \mathbf{r}_a^{(j_a)}\} \tag{11.107}$$

其中，每一个用户兴趣表示都会寻找与其最匹配的物品属性表示，并累积所有兴趣维度上的匹配结果。相比传统双塔模型单一向量匹配方式，该机制能够更充分地捕获用户多兴趣与物品多属性之间的复杂关系。同时，由于物品侧多头表示可以提前离线计算并缓存，因此 HIT 模型仍然保留了双塔模型高效在线推理的特点，不需要像全交互模型比如 COLD、FSCD 模型一样去在线计算完整的用户与物品交叉网络。

11.5.2.3 HIT 模型优化目标

为了同时优化预测能力以及生成表示的质量，HIT 模型采用交叉熵损失和生成损失联合训练。其中，交叉熵损失用于优化最终排序预测结果：

$$L_c = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\sigma(\hat{y}_i)) + (1 - y_i)\log(1 - \sigma(\hat{y}_i))) \tag{11.108}$$

此外，为了约束 Generator 生成的模拟表示能够逼近真实的对侧表示，HIT 模型进一步设计生成损失。该损失通过余弦距离衡量 Generator 输出表示与多头表示器（Multi-head Representer）输出表示之间的差异：

$$L_g = -\frac{1}{N}\sum_{i=1}^{N}(y_i Dist(\mathbf{r}_u, \mathbf{r}_m^1) + (1 - y_i)Dist(\mathbf{r}_u, \mathbf{r}_m^2)) \tag{11.109}$$

其中，$Dist(\cdot)$ 表示余弦距离。最终模型优化目标为：

$$L = L_c + \alpha(L_{g_\mathbf{u}} + L_{g_\mathbf{a}}) \tag{11.110}$$

其中，α 用于控制生成损失的权重。通过联合优化预测任务与生成任务，HIT 模型能够同时学习用户-物品匹配关系以及跨塔语义关联关系。

整体来看，HIT 模型代表了工业界粗排双塔模型进一步增强交互能力的一种探索方向。相比传统双塔模型，HIT 模型通过生成式跨塔信息融合以及多头兴趣匹配机制，有效缓解了用户与物品表示完全独立带来的信息缺失问题；相比 COLD 等全交互模型，HIT 模型仍然保持了双塔模型 Embedding 预计算和低延迟推理优势。因此，该模型能够较好地平衡粗排阶段对于模型效果和在线效率的双重需求，并已经在腾讯广告推荐系统中进行了工业化部署。

11.6 粗排模型实践案例

前面章节主要介绍了工业级推荐系统粗排阶段中的典型模型结构，包括双塔模型、多塔模型、粗排 DNN 模型、多兴趣建模方法以及蒸馏优化方法等。然而，在真实工业推荐系统中，粗排模型通常并不会直接采用某一种固定的网络结构，而是会根据具体业务场景、用户行为特点以及线上计算资源约束进行持续迭代优化。

从整体发展趋势来看，工业界粗排模型通常以双塔模型作为基础 Backbone，通过不断增加新的信息建模模块来提升模型表达能力。例如，早期粗排模型主要依赖用户塔和物品塔独立编码，通过向量匹配完成候选排序；

随着业务复杂度提升，算法工程师会逐步在双塔结构基础上引入显式特征交叉模块、用户兴趣建模模块、多模态表征模块以及复杂的多任务学习结构，使粗排模型逐渐具备接近精排模型的预测能力。

需要注意的是，粗排模型与精排模型最大的区别仍然在于计算效率约束。由于粗排阶段需要处理数千甚至上万个候选物品，因此任何模型结构的增加都需要充分考虑线上推理成本。因此，工业级粗排模型的优化通常不是简单追求模型复杂度，而是在模型表达能力、在线延迟、计算资源以及业务收益之间寻找平衡。

除此之外，随着推荐系统业务目标不断丰富，粗排模型通常也会采用多任务学习（Multi-task Learning）的方式同时优化多个业务目标。以短视频推荐中的上下滑场景为例，用户对于推荐内容的反馈并不仅仅包括点击行为，还包括有效播放、观看时长、长播、点赞、评论、收藏以及分享等多个维度。不同用户行为反馈反映了用户兴趣的不同侧面，因此粗排模型通常需要同时预测多个目标概率，从而更加全面地刻画用户对于候选物品的综合偏好。

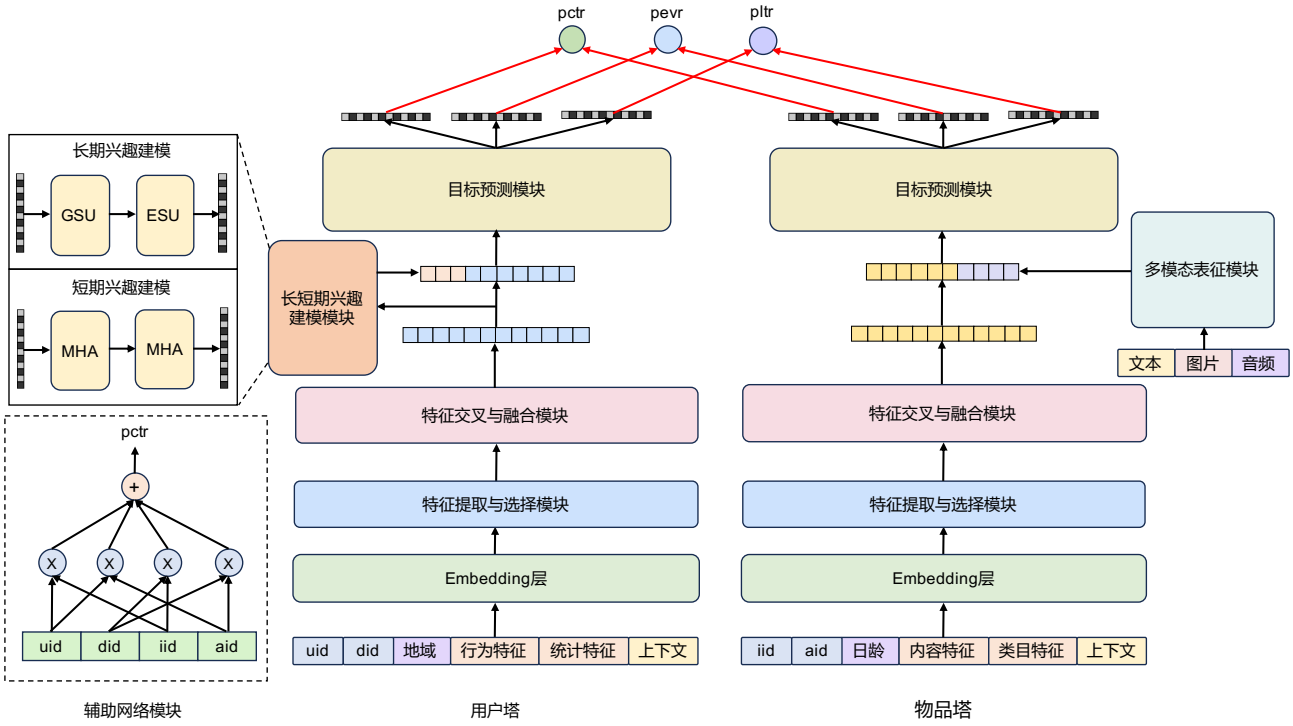

图 11.23: 典型工业粗排模型网络结构示意图。

典型的工业粗排模型网络结构如图 11.23所示，整体通常包括特征提取与选择模块、特征交叉与融合模块、长短期兴趣建模模块、多模态表征模块、目标预测模块以及辅助网络模块等多个组成部分。不同模块分别负责从不同角度增强粗排模型的表达能力。

特征提取与选择模块主要负责对用户侧和物品侧原始特征进行 Embedding 编码，将离散 ID 特征、统计特征以及上下文特征转换为低维稠密向量表示。在工业场景中，由于输入特征规模通常非常庞大，不同特征对于最终预测目标的重要性存在明显差异，因此通常还会进一步引入特征选择机制。例如，可以通过 LHUC（Learning Hidden Unit Contribution）、SE-Net、IntTower 中的 Light-SE 模块等结构，对不同特征 Embedding 进行动态加权，从而突出更加重要的特征信息，降低无效特征对于模型训练的干扰。

特征交叉与融合模块主要用于建模用户特征与物品特征之间，以及不同用户特征之间的高阶交互关系。相比传统双塔模型仅通过 Embedding 内积完成匹配，工业粗排模型通常会进一步引入特征交叉网络增强模型表达能力。例如，可以采用 FM、DeepFM 等方法建模低阶和高阶特征交互，也可以采用 DCN 等显式交叉网络提取组合特征。此外，随着 Transformer 结构在推荐系统中的广泛应用，越来越多粗排模型开始采用基于注意力机制的特征交叉模块，通过 Self-Attention 学习不同特征之间更加复杂的关联关系。

长短期兴趣建模模块主要用于捕获用户动态变化的兴趣特征。用户行为序列通常包含明显的时间演化规律，近期行为更多反映用户当前兴趣，而长期行为则体现用户稳定偏好。因此，工业粗排模型通常会同时建模短期兴

趣和长期兴趣。对于短期兴趣建模，例如用户最近几十次观看、点击、有效播放、点赞、评论以及分享行为序列，通常会采用 Transformer 等序列建模方法进行编码，以捕获用户近期兴趣变化趋势。在具体实现过程中，通常会将当前用户 Embedding 作为 Query，对用户历史行为序列进行注意力检索，从而得到与当前推荐场景最相关的短期兴趣表示。对于长期兴趣建模，由于用户历史行为可能达到数月甚至数年的规模，直接输入完整行为序列会带来巨大的计算开销，因此工业系统通常采用 SIM 等长期兴趣检索方法。具体来说，会根据不同 Trigger（例如点击、购买、长播、点赞等行为类型）从用户长期历史行为中检索 Top-K 相关行为序列，再通过 Transformer 等序列模型进行进一步编码，最终得到多个长期兴趣表示。随后，将长短期兴趣 Embedding 与主网络 Embedding 进行 Concat 融合，共同输入后续预测网络。

需要注意的是，由于传统双塔模型中的物品塔通常采用离线预计算方式，因此在长短期兴趣建模过程中，一般主要使用用户塔 Embedding 作为 Query 进行兴趣检索，而不会直接使用物品塔 Embedding 参与在线兴趣搜索。只有在引入 DAT 等跨塔信息融合机制后，物品塔表示包含用户侧相关信息时，才可能进一步利用物品 Embedding 参与兴趣检索。

多模态表征模块主要用于融合用户和物品的多模态信息，包括文本、图像、视频、音频等内容特征。随着推荐系统中内容信息的重要性不断提升，仅依赖 ID 特征已经难以充分描述物品语义，因此越来越多粗排模型开始引入多模态 Embedding 作为额外信息来源。

在实际工业部署中，多模态信息融合方式具有不同复杂程度。较简单的方法通常直接将预训练得到的多模态 Embedding 作为输入特征，与原始 Embedding 进行 Concat 融合，然后通过 DNN 网络进行预测。更加复杂的方法则会进一步采用 Transformer 等结构，对不同模态之间的信息进行交叉建模，从而获得更加丰富的语义表示。关于多模态推荐中的具体方法将在后续第34章中进行详细介绍。

目标预测模块主要负责根据融合后的用户和物品表示，对多个业务目标进行预测。由于粗排模型通常需要同时优化点击率、播放时长、转化率等多个目标，因此工业系统通常会采用多任务学习结构，例如 MMOE、CGC、PLE 等网络结构，通过共享底层信息以及任务特定网络，提高多个目标之间的联合建模能力。当然，在业务目标较少或者模型复杂度要求较高的情况下，也可以采用简单的 DNN 网络分别预测不同目标。

辅助网络模块主要用于提供额外监督信号，帮助主网络学习更加有效的特征表示。由于辅助网络通常不参与线上推理，因此其结构设计可以更加灵活。例如，可以采用 FM、浅层 DNN 等轻量网络预测核心业务目标，使共享 Embedding 特征得到更加充分的训练；也可以借鉴粗排蒸馏模型中的 Teacher 网络，通过更强大的辅助网络为主模型提供额外知识监督，从而提升粗排模型最终效果。

整体来看，工业级粗排模型实际上是在双塔高效匹配框架基础上，通过不断引入新的信息来源和模型结构，实现从简单向量匹配向复杂用户-物品关系建模的逐步演进。但与此同时，所有结构优化都需要充分考虑线上计算效率，这也是粗排模型区别于精排模型最核心的工程约束。

11.7 粗排模型实践讨论

在工业级推荐系统的实际优化过程中，粗排模型的迭代通常经历从模型结构优化、目标扩展优化，到链路一致性优化的发展过程。由于粗排模型处于推荐链路的中间阶段，其核心目标是在严格的在线延迟约束下尽可能提升候选集筛选的准确性。因此，粗排模型的优化并不是单纯追求模型复杂度提升，而是在计算成本、模型表达能力以及推荐链路整体收益之间寻找平衡。

在粗排模型发展的早期阶段，算法工程师通常主要通过网络结构改进来提升模型的表达能力。例如，在基础双塔模型的基础上，引入显式特征交叉模块、长短期兴趣建模模块、多模态表征模块以及更加复杂的多任务学习结构等。由于早期粗排模型通常采用较为简单的 Embedding 拼接和 DNN 预测结构，因此引入 Transformer、MMOE、SIM 等复杂网络结构往往能够带来较为明显的线上收益。同时，在网络结构升级过程中，通常也会伴随着新的用户特征、物品特征以及上下文特征的引入，使模型能够更加充分地理解用户兴趣和物品属性。

然而，随着粗排模型经过多轮迭代优化，其线上 baseline 不断提升，单纯依靠网络结构升级所带来的收益会逐渐降低。一方面，复杂网络结构虽然能够提升模型的表达能力，但同时也会增加模型参数规模、训练成本以及线上推理延迟；另一方面，当模型已经能够较充分地建模已有特征信息时，继续增加网络深度或者引入更加

复杂的交互结构，能够挖掘的信息增量也会越来越有限。因此，在工业实践中，粗排模型优化通常会逐渐从单纯的网络结构优化转向更加关注业务目标和数据增量的信息挖掘。

其中，一个非常常见且有效的优化方向就是所谓的“加 label”方案。所谓加 label，并不是简单地增加一个预测目标，而是根据具体业务场景中用户行为反馈的特点，设计新的监督信号或者 reward 目标，并通过新增预测 Tower 将这些额外的信息引入粗排模型。例如，在短视频推荐场景中，除了传统的点击率目标之外，还可以进一步引入有效播放、长播放、点赞、评论、分享、关注等行为目标；在直播推荐场景中，则可以进一步引入进入直播间、停留时长、互动行为以及长期价值相关目标。通过增加新的行为反馈目标，模型能够学习到更加丰富的用户兴趣表达，从而提升粗排阶段对于候选物品价值的判断能力。

从本质上来看，“加 label”优化的核心并不是增加模型结构，而是通过引入新的监督信息扩展模型能够学习的用户行为空间。因此，在实际工业推荐系统中，一个好的 label 设计往往比单纯调整网络结构更加重要。例如，新增一个能够反映用户长期价值的目标，可能比增加更多复杂网络层带来的收益更加显著。

但是，加 label 并不是一种可以无限扩展的优化方式。在多目标粗排模型的发展过程中，如果算法工程师不断增加新的预测目标，通常会导致模型结构越来越复杂。例如，每增加一个业务目标，往往都需要对应增加一个目标预测 Tower，从而导致模型参数规模快速增长。同时，大量 Tower 之间可能存在较强的语义相关性，例如点击、有效播放、长播放本质上都反映用户消费兴趣，而点赞、评论、分享等行为更多反映用户主动互动意愿。如果完全采用独立 Tower 进行建模，不仅会造成参数冗余，也会增加训练过程中的显存占用和计算开销，甚至导致模型训练过程中出现 OOM 等工程问题。

因此，在多目标粗排模型优化过程中，通常需要进一步进行 merge tower 处理，即对多个目标预测网络进行合理的参数共享和结构融合。具体而言，可以根据不同目标之间的行为语义关系设计共享网络结构。例如，有效播放、长播放、点击等消费类目标通常具有较强相关性，可以共享部分底层网络参数；点赞、评论、分享等互动类目标也可以共享部分网络层，同时保留目标特定的预测层。通过这种方式，可以在保持多目标预测能力的同时，有效降低模型规模，提高训练和部署效率。

当粗排模型经过多轮结构优化和目标扩展之后，进一步提升模型效果的关键方向通常会逐渐转向推荐链路的一致性优化。在工业推荐系统中，粗排模型和精排模型实际上服务于同一个推荐目标，但由于二者所处阶段、候选规模以及模型复杂度存在明显差异，因此优化目标往往存在一定偏差。例如，粗排模型更加关注在大规模候选集合上的快速筛选能力，而精排模型则能够利用更加丰富的特征和更加复杂的网络结构进行精细化排序。如果粗排模型与精排模型优化目标不一致，则可能导致粗排阶段提前过滤掉大量精排模型认为具有价值的候选物品，从而影响最终推荐效果。

因此，一些常见的优化迭代思路就逐渐转变为粗排模型与链路中下游精排模型的联动，即粗排链路一致性的优化。在这种优化思想下，粗排模型的迭代通常不再只是单纯考虑粗排模型自身对于各种 label 预估精度的提升，而是需要进一步考虑粗排模型在整个推荐链路中的具体生效情况，以及粗排模型与精排模型之间的目标一致性。

具体而言，一方面，可以通过知识蒸馏方法，将精排模型中的排序知识迁移到粗排模型中，使粗排模型学习精排模型对于候选物品价值的判断能力；另一方面，也可以通过候选集合一致性优化，使粗排模型输出的 TopK 候选集合更加接近精排模型最终选择的候选集合。通过这种方式，粗排模型能够利用精排模型的预估结果、排序信息以及候选选择行为等信息进行学习，从而使粗排模型和精排模型之间更加一致，并进一步提升整个推荐链路的最终收益。这部分链路一致性优化的内容，我们将在后续第21章节中进行详细介绍。

除了链路一致性优化之外，近年来随着大语言模型（Large Language Model, LLM）和 AI 技术的快速发展，粗排模型的优化迭代也开始探索如何将大语言模型以及相关 AI 技术引入到粗排模型中，从而进一步提升模型的表达能力和预测性能。例如，可以在粗排模型中引入大语言模型生成的 Embedding 作为特征输入，用于增强用户兴趣理解和物品语义表示能力；也可以利用大语言模型对用户历史行为进行兴趣抽象，对物品文本、知识属性等多维信息进行理解，从而提升粗排模型对于复杂用户兴趣和物品属性的刻画能力。

除了引入 LLM 相关 Embedding 之外，未来粗排模型的发展也会逐渐借鉴大模型领域的 scaling law 思想，即通过增加模型参数规模、训练数据规模以及计算资源投入，进一步提升模型表达能力和预测性能。传统粗排模型由于受到在线延迟约束，通常更加关注轻量化设计，但随着模型压缩、Embedding 缓存、硬件加速以及推理优

化技术的发展，粗排模型也具备进一步扩大模型规模的可能性。关于大语言模型、Agent 技术如何赋能推荐系统，以及 LLM 与推荐模型结合的发展方向，我们将在后续第37章节中进行详细介绍；关于推荐大模型的发展、模型规模扩展以及基于 scaling law 思想构建下一代推荐模型的内容，我们将在后续第36章节中进行详细介绍。

总体来看，工业级粗排模型的发展过程，本质上是从简单的双塔匹配模型，逐渐演进为融合复杂特征交互、多兴趣建模、多目标学习、链路一致性优化以及大模型能力的新型推荐基础模型。未来粗排模型的优化重点，也将不仅仅局限于模型结构本身，而会更加关注如何利用更丰富的信息、更强的模型能力以及更合理的推荐链路协同机制，进一步提升整个推荐系统的整体收益。

11.8 本章小结

在本章节中，我们围绕工业级推荐系统中的粗排模型进行了系统性的介绍。作为连接召回阶段与精排阶段的重要中间环节，粗排模型需要在严格的在线延迟约束下，对大规模候选集合进行高效筛选，因此其核心目标是在模型表达能力和线上计算效率之间取得平衡。本章节从经典模型范式、工业优化方向以及实际工程实践三个角度，对粗排模型的发展历程和关键技术进行了详细分析。

首先，我们介绍了粗排模型中最经典的双塔模型范式。双塔模型通过用户塔和物品塔分别进行特征编码，并通过向量匹配完成用户与物品之间的相关性计算，是工业推荐系统中应用最广泛的粗排架构之一。本章节首先介绍了最早提出的 DSSM 模型，并进一步介绍了基于 DSSM 框架演进出的多种改进方法，包括融合 Transformer 序列建模能力的 DSSM + Transformer 模型、腾讯提出的 TDSSM 模型以及通过跨塔信息交互增强双塔表达能力的 DAT 模型。通过这些模型的介绍，我们分析了双塔模型从最初的独立表示学习，到逐步引入用户-物品交互信息的发展过程。

除此之外，我们进一步讨论了双塔模型天然存在的问题，即由于用户塔和物品塔采用独立编码方式，用户侧和物品侧特征之间缺少充分交互，导致模型对于复杂用户-物品关系的刻画能力有限。因此，本章节进一步介绍了多塔模型这一类增强双塔交互能力的方法。具体而言，我们介绍了 MV-DSSM 模型、DSSM + FM 模型以及延迟反馈多塔模型等经典工作。其中，在延迟反馈多塔模型部分，我们重点分析了工业推荐系统中实时数据流和离线数据流并存情况下的建模方式，讨论了如何通过不同数据流分别训练不同目标模型，从而解决即时反馈和延迟反馈目标之间的优化冲突问题。最后，我们进一步介绍了引入显式交叉特征的多塔模型，展示了工业界如何在保持一定在线效率优势的同时，增强用户与物品之间的信息交互能力。

随后，我们对粗排模型中的知识蒸馏方法进行了系统介绍。由于精排模型通常具有更强的表达能力，但受到在线延迟限制无法直接应用于大规模候选筛选，因此知识蒸馏成为提升粗排模型性能的重要技术路线。本章节首先介绍了经典的 Teacher-Student 知识蒸馏框架，并重点分析了显式知识蒸馏和隐式知识蒸馏两种主要方式。其中，显式知识蒸馏通过迁移 Teacher 模型输出的预测分布或者排序结果进行监督，而隐式知识蒸馏则通过迁移中间层表示、特征关系等方式增强 Student 模型的表达能力。在此基础上，我们进一步介绍了工业界具有代表性的粗排知识蒸馏模型，包括 Rocket Launching 模型和 PFD 模型，分析了这些方法如何利用更强大的 Teacher 网络辅助轻量级粗排模型学习。

接下来，我们重点介绍了粗排 DNN 模型的发展方向。随着工业推荐系统中用户行为、物品属性以及上下文信息不断丰富，仅依赖传统双塔结构已经难以满足复杂业务场景的建模需求。因此，工业界逐渐探索在粗排模型中引入更加灵活的深度神经网络结构。本章节首先介绍了基于双塔结构的双塔 DNN 模型，即在用户塔和物品塔顶部 Embedding 表示基础上进一步引入 DNN 网络进行特征融合。同时，我们也介绍了完全基于全连接 DNN 结构的粗排模型架构，包括工业界经典的 COLD 模型、FSCD 模型以及 IntTower 模型。其中，IntTower 通过轻量级特征交互机制，在保持双塔高效推理能力的同时增强用户-物品之间的细粒度交互能力，体现了工业界对于“效果提升”和“在线效率”平衡的探索。

在此基础上，我们进一步讨论了粗排模型中的多兴趣建模问题。由于真实用户通常具有多样化且动态变化的兴趣，仅使用单一用户 Embedding 难以完整表达用户复杂偏好，容易导致不同兴趣之间的信息混淆。因此，在粗排模型中引入多兴趣建模成为提升模型效果的重要方向。本章节介绍了两个具有代表性的工业模型，即腾讯提出的 MVKE 模型和 HIT 模型。其中，MVKE 通过 Virtual Kernel Expert 机制学习用户不同兴趣空间，并利用

Virtual Kernel Gate 动态组合不同兴趣表示；HIT 模型则进一步结合跨塔信息交互和多头表示机制，在保持双塔高效推理能力的同时，实现更加细粒度的用户兴趣和物品属性匹配。

最后，我们结合前面介绍的不同粗排模型技术方向，对工业级推荐系统中的粗排模型实践进行了总结和分析。实际工业系统中的粗排模型并不是简单采用某一种固定架构，而通常是由双塔基础结构、多塔网络、DNN 交互模块、多兴趣建模模块、多任务预测网络、知识蒸馏模块以及各种业务辅助网络共同组成的复杂系统。我们进一步介绍了粗排模型在工业实践中的优化演进过程，包括早期通过网络结构升级和特征扩展提升模型表达能力，到通过新增业务 Label 引入更多用户反馈信息，再到通过 Merge Tower 降低多目标模型复杂度，以及进一步通过粗排-精排链路一致性优化提升整个推荐链路收益。

此外，本章节也讨论了近年来大模型和 AI 技术发展对于粗排模型未来演进方向的影响，包括利用 LLM 生成的 Embedding 增强用户和物品语义理解能力，以及借鉴大模型领域 Scaling Law 思想，通过扩大模型规模和计算能力提升粗排模型表达能力。通过这些内容的介绍，我们希望读者能够不仅理解各种经典粗排模型的结构设计，更能够从工业推荐系统优化的视角理解粗排模型为什么需要不断演进，以及不同技术方向在真实业务场景中的应用价值。

总体而言，本章节系统梳理了工业级推荐系统粗排模型的发展路径。从最初追求高效率的双塔匹配模型，到逐渐引入跨塔交互、多任务学习、知识蒸馏、多兴趣建模以及大模型能力增强，粗排模型的发展过程本质上体现了推荐系统在大规模工业部署条件下，不断探索模型效果、计算效率和业务收益之间平衡的过程。

在完整的推荐系统链路中，粗排模型主要承担从海量候选集合中快速筛选高价值候选物品的任务，其优化目标更加关注在有限计算资源和严格延迟约束下提升候选集质量。然而，由于粗排阶段仍然面临候选规模较大、线上计算资源有限等限制，粗排模型通常无法使用过于复杂的网络结构对用户和物品之间的关系进行充分建模。因此，在粗排模型完成候选集压缩之后，推荐系统还需要通过更加精细化的排序模型进一步学习用户与物品之间的复杂交互关系，对候选物品进行更加准确的价值评估。

下一章节我们将进一步介绍工业级推荐系统中的精排模型（Ranking Model）。相比于粗排模型，精排模型面对的候选集合规模更小，因此可以采用更加复杂的特征交互结构、更丰富的用户行为建模方式以及更加精细化的多目标优化方法。本章节提到的许多粗排模型技术，例如多任务学习、多兴趣建模、知识蒸馏以及链路一致性优化，也会进一步与精排模型产生紧密联系。下一章节将围绕精排模型的发展历程、经典网络结构、工业实践方法以及其他精排优化方向展开详细的介绍。

11.9 参考文献

[1] Olivier Chapelle. “Modeling delayed feedback in display advertising”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 1097–1105.

[2] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. “A multi-view deep learning approach for cross domain user modeling in recommendation systems”. In: *Proceedings of the 24th international conference on world wide web*. 2015, pp. 278–288.

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[4] Po-Sen Huang et al. “Learning deep structured semantic models for web search using clickthrough data”. In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2013, pp. 2333–2338.

[5] Sofia Ira Ktena et al. “Addressing delayed feedback for continuous training with neural networks in CTR prediction”. In: *Proceedings of the 13th ACM conference on recommender systems*. 2019, pp. 187–195.

[6] Xiangyang Li et al. “Inttower: the next generation of two-tower model for pre-ranking system”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, pp. 3292–3301.

[7] Xu Ma et al. “Towards a better tradeoff between effectiveness and efficiency in pre-ranking: A learnable feature selection based approach”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, pp. 2036–2040.

[8] Qi Pi et al. “Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2685–2692.

[9] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. “Multi-rate deep learning for temporal recommendation”. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2016, pp. 909–912.

[10] Yanshi Wang et al. “Delayed feedback modeling for the entire space conversion rate prediction”. In: *arXiv preprint arXiv:2011.11826* (2020).

[11] Zhe Wang et al. “Cold: Towards the next generation of pre-ranking system”. In: *arXiv preprint arXiv:2007.16122* (2020).

[12] Chen Xu et al. “Privileged features distillation at taobao recommendations”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 2590–2598.

[13] Zhenhui Xu et al. “Mixture of virtual-kernel experts for multi-objective user profile modeling”. In: *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2022, pp. 4257–4267.

[14] Haoqiang Yang et al. “HIT model: A hierarchical interaction-enhanced two-tower model for pre-ranking systems”. In: *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 2025, pp. 6201–6208.

[15] Yantao Yu et al. “A dual augmented two-tower model for online large-scale recommendation”. In: *DLP-KDD* (2021).

[16] Guorui Zhou et al. “Rocket launching: A universal and efficient framework for training well-performing light net”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.