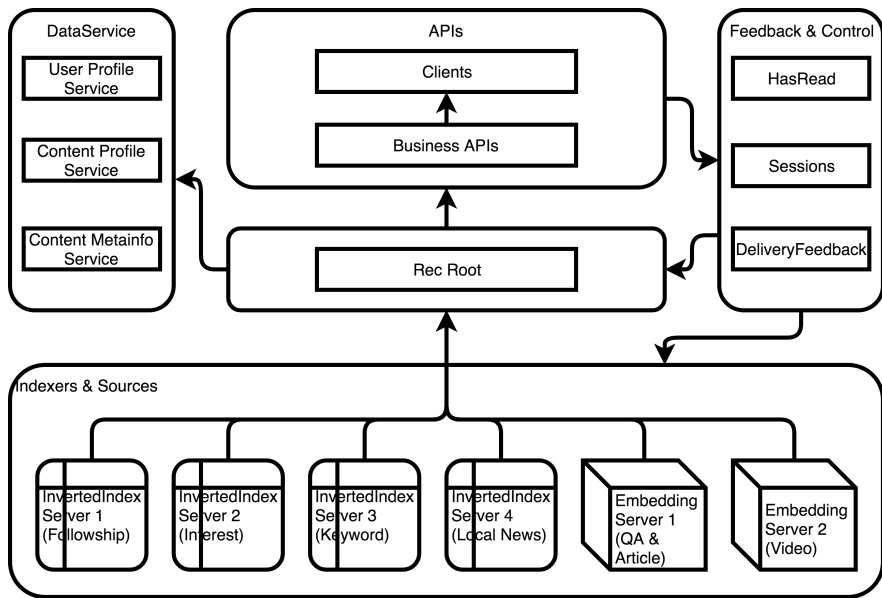


机器学习在知乎 Feed 流推荐系统中的应用

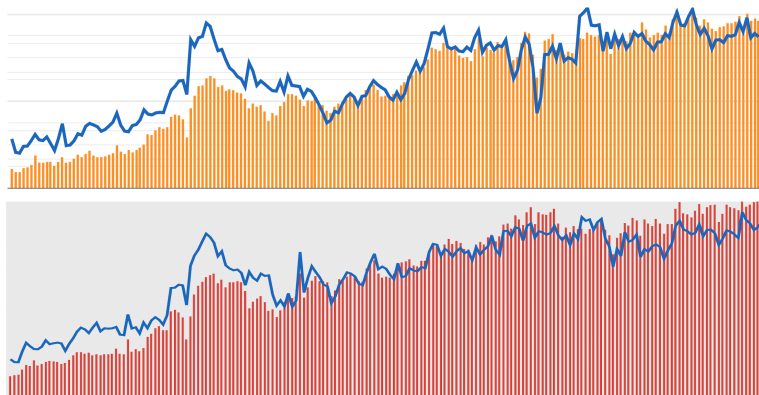
知乎首页业务总监 张瑞

知乎信息流推荐系统简介

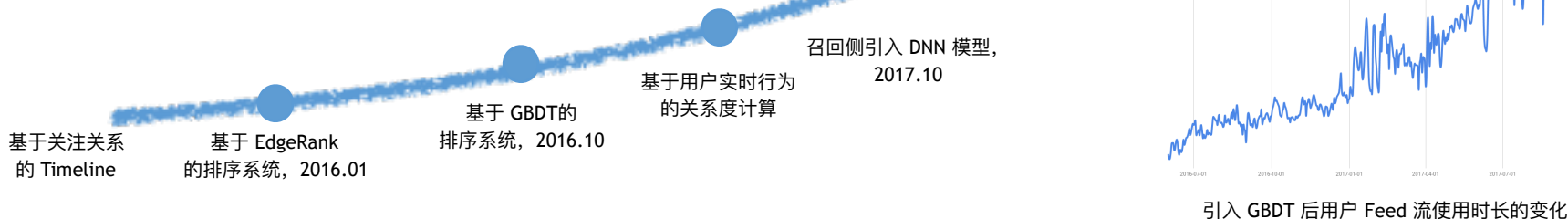
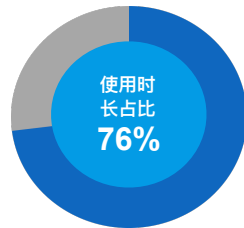
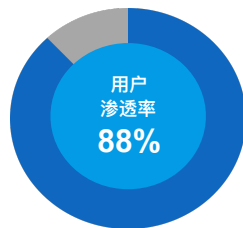
水晶球：基于多策略融合的多源内容推荐系统



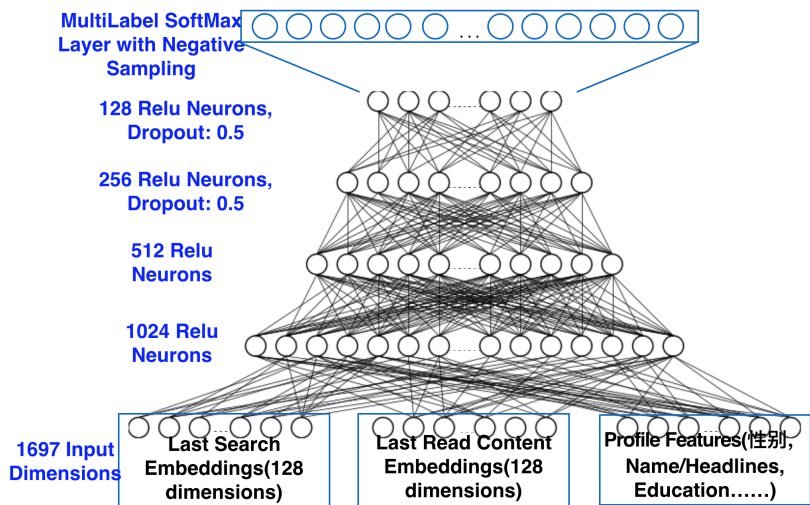
知乎信息流推荐系统演进



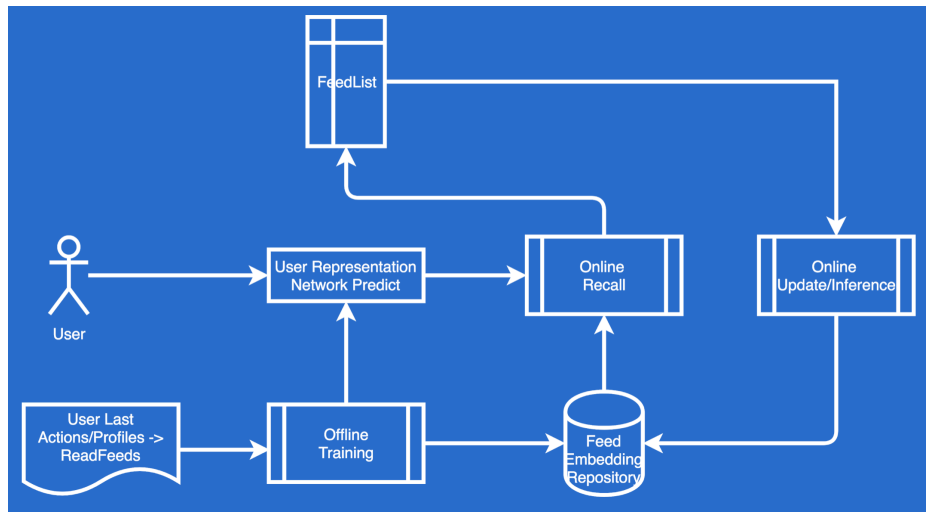
引入 DNN 召回及排序后用户 Feed 流使用时长和阅读量的变化



基于深度学习的推荐召回 - v1.0

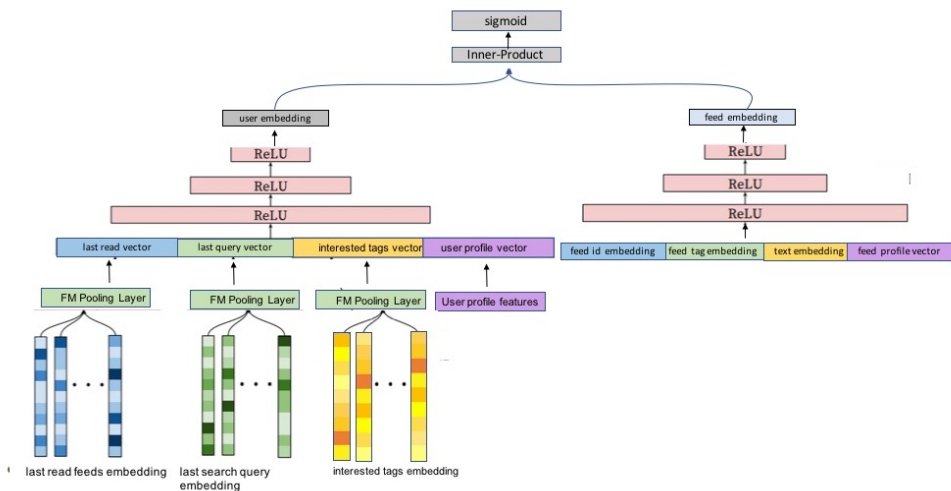


- 训练样本采集：从用户行为序列中收集用户在阅读某 Feed 时候的 Profile(历史/实时) + Context(时间窗口内的搜索词/阅读历史)
- 对 Feed 按频次过滤，滤掉在训练集中出现次数小于一定阈值的 Feed
- 将用户的 Profile、历史行为表示成一组 128 维向量
- 使用 Faiss 作为向量化 ANN 召回的 Backend



- 训练过程中生成一批 Feed Embedding
- 在线使用 User Representation Network 生成的 User Embedding，从 Feed Embedding 中检索最匹配的结果召回
- 线上使用 spark streaming, batch update 新产生的 Feed 的 Embedding 及既有内容的 Embedding

基于深度学习的推荐召回 - v2.0

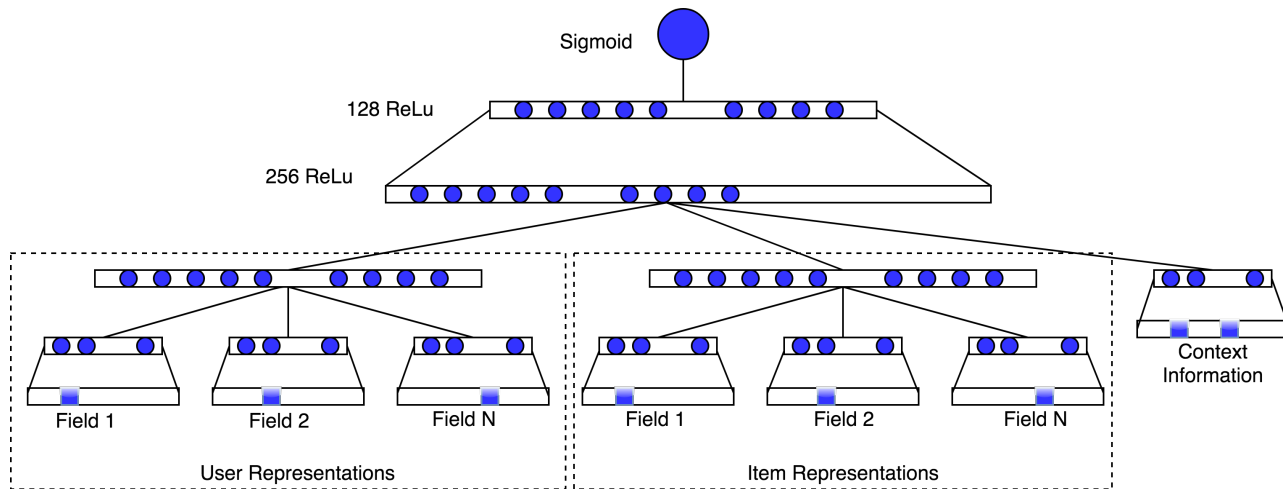


- User Representation Network 引入 FM Pooling 层，学习用户高频消费行为的交叉特征
- Top 100 ACC +8%

	原始网络	Top 2w QueryTag ID 化	Top 2w Query Tag ID 化+Top 2w 文章 Tag ID 化
Average	0.168	0.189	0.196
FMPooling	-	0.195	0.210

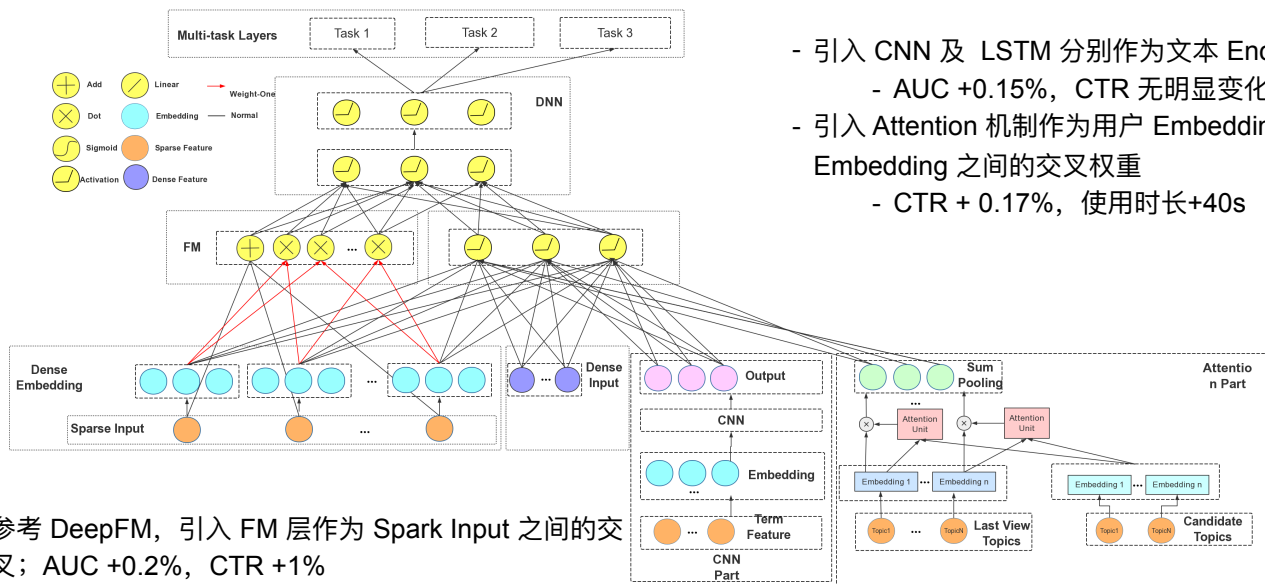
- 训练 Content Embedding Network，使用 Content 原始特征直接得到 Content Embedding，解决新内容的召回机制问题
- 利用「展示未点击的 Skip 数据」
 - 使用 Skip 数据作为特征
 - 使用 Skip 数据指导采样

基于深度学习的 CTR 预估模型 - v1.0



- 总体结构和基于 DNN 的召回框架类似；使用两层 Relu 而不是直接点积作为 Embedding 的预估网络
- 引入 Context Information 作为特征：时间段、网络、机型、location 信息，等等
- 输入层细分为 11 个 Block，减少特征在初始出入的错误交叉，减轻过拟合；AUC +1%，CTR +5.8%
- Wide & Deep Network：无明显提升；Wide 部分需要比较强的特征工程

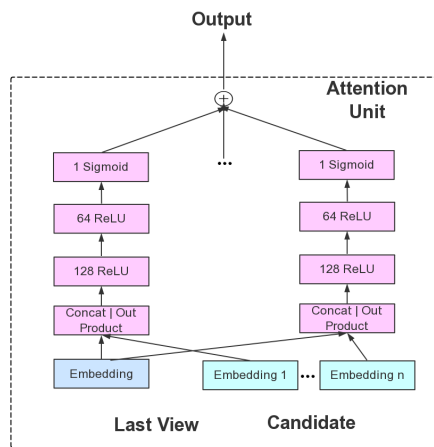
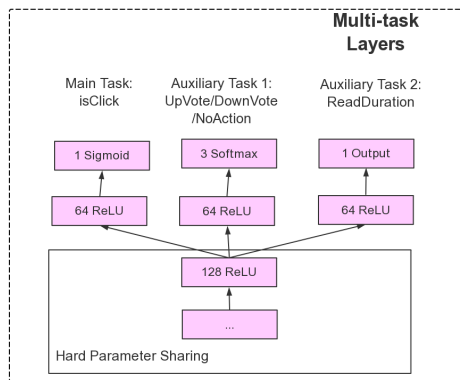
基于深度学习的 CTR 预估模型 - v2.0



- 引入 CNN 及 LSTM 分别作为文本 Encoder/Last Action Encoder
 - AUC +0.15%，CTR 无明显变化，使用时长 +50s
- 引入 Attention 机制作为用户 Embedding 和 Candidate Embedding 之间的交叉权重
 - CTR + 0.17%，使用时长+40s

- 参考 DeepFM，引入 FM 层作为 Sparse Input 之间的交叉；AUC +0.2%，CTR +1%

基于深度学习的 CTR 预估模型 - 多目标预估



- 对于社区化的平台，点击（阅读）不是唯一的业务指标
 - 需要预估阅读时长、点赞、收藏、分享、创作等行为
- 解决多目标预估中的效率问题
 - 使用 CTR 预估模型预训练网络；利用 Parameter Hard Sharing，降低前向/反向传播中的计算量
- 对于非离散特征如阅读时长，进行 Normalize 操作
 - DwellTime Normalize:
$$\hat{z}_i = \frac{\log(t_i) - \mu_{C_i}}{\sigma_{C_i}}$$
 - 修改 Softmax 函数预估 z-value

推荐系统中的实际问题

- 模型训练问题

- 样本组织

- 线上实时打点避免特征穿越

- 对于推荐系统，正向行为通常远比负向行为稀疏；正负样本进行不同采样率的实验

- 特征工程

- 对于分布范围较大的特征，进行特征归一化和 boxing

- 模型评估

- AUC 是基础指标，DCG Gain 具有更高的参考意义： $DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$ DCG Gain: $DCG_{10, reranked} / DCG_{10, online} - 1.0$

- 业务问题：Is Accuracy All We Need?

- 多样性问题如何解决？

- 健壮的 Rerank 框架，进行业务导向的调权、打散、隔离和禁闭

- 如何避免「信息茧房」的产生？

- Explore & Exploit 机制

- E&E 过程利用 Tag 之间的关联信息，增强效率

Q&A