

深度学习在美团点评推荐业务中实践

前言

近年来，深度学习在语音识别、计算机视觉及自然语言处理等领域都取得了很大的突破，成为学术界和工业界关注的热点。与传统机器学习方法相比，深度学习在特征抽取及特征组合方面具有明显的优势，可以学习到多层次的抽象特征表示，为复杂的非线性系统提供优秀的建模能力。美团点评，作为生活服务平台，有数亿的用户及丰富的用户行为，在线上与线下相结合的场景下，用户的个性化需求越来越多，推荐系统变得尤为重要。在这种背景下，将深度学习算法应用到推荐业务中，改进并优化目前的推荐算法，使得推荐效果更为智能化，用户体验更好变得非常重要。本文将结合具体的业务场景，介绍深度学习在美团点评推荐上的实践经验及一些思考。

点评推荐平台概述

美团点评，由于自身业务丰富，且用户的消费场景多变，我们推荐的场景也需要随着用户兴趣、地点、环境、时间等变化而变化。为了能够为用户提供好的信息发现体验，支撑业务的快速发展，点评推荐平台面临着以下新的挑战：

- 业务与场景信息丰富：除了传统的类电商的团购、闪惠、酒店预订等业务，还有外卖这样的到家消费业务。同时，用户的场景信息多变，例如用户地理位置在变：用户可以在家、在商场或者在门店，不同场景下用户的兴趣差别很大；气候环境在变：例如大雨天气或者雾霾等环境信息会影响用户兴趣。
- 内容化推荐：内容已经成为互联网近几年发展的重点，除了点评的 UGC 内容，我们可以看到头条、视频、“探店报告”等不同内容形式，也可以看到一些内容导流型产品，例如“特色推荐菜”等以 SKU 为维度的内容，“好友热搜”等围绕一个主题组织的排行榜形式内容。不同的内容形态，对推荐系统的要求差异很大。

针对以上的问题，我们开发了可以适配不同业务形态、方便业务快速接入的推荐平台，包括多策略选品的召回及机器学习排序框架，从而向用户推荐感兴趣的信息，让用户感觉到欣喜。整个推荐平台包括离线的海量数据挖掘、近线的实时意图预测以及在线的高并发服务。推荐平台的策略主要分为召回和排序两个过程，召回主要负责生成推荐的候选集，排序负责将结果进行个性化的精准排序。点评的推荐平台业务架构及服务架构如下图所示：



图1 点评推荐平台业务架构

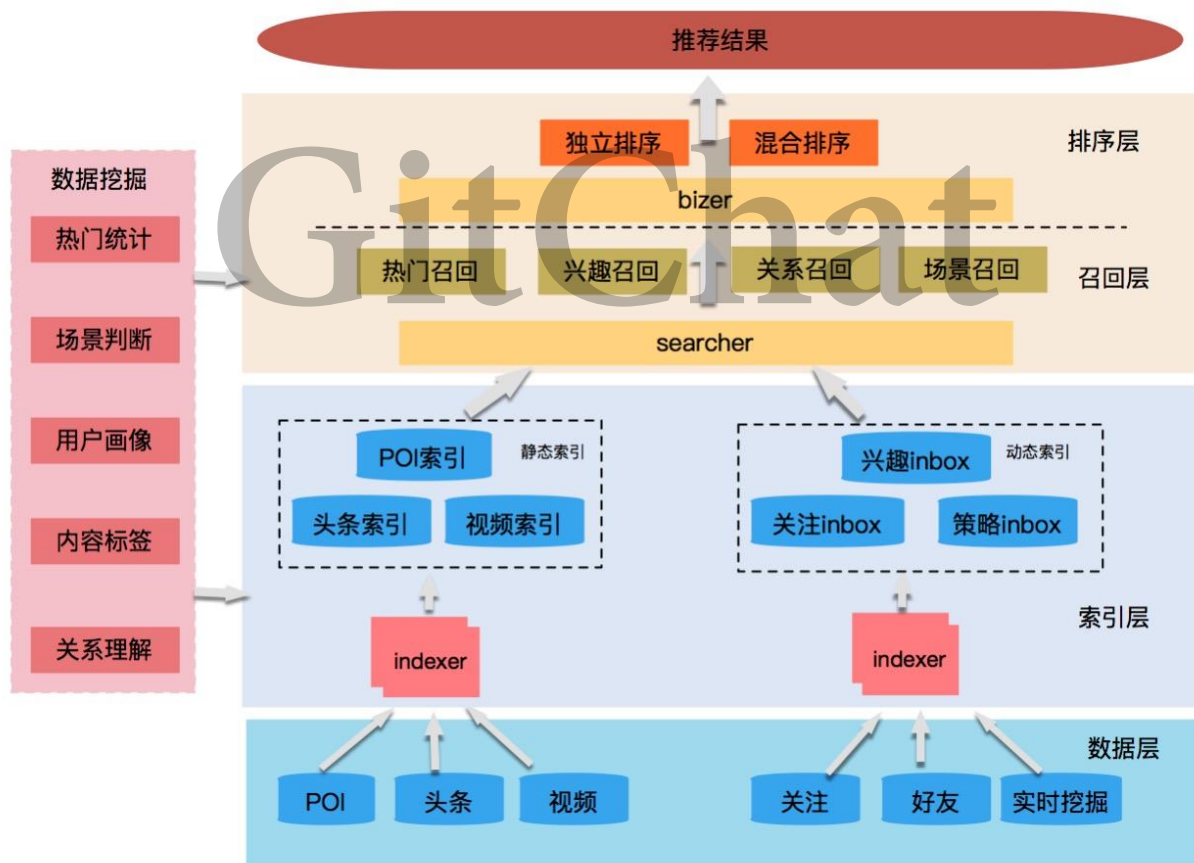


图2 点评推荐平台服务架构

深度学习在推荐中的应用

在推荐平台的构建过程中，多策略选品和排序是两个非常重要的部分，本文接下来主要介绍深度学习相关的推荐算法，主要包括 DSSM、Session Based RNN 推荐召回模型与 Wide Deep Learning 的排序模型，我们会介绍深度学习模型在推荐业务应用及实现的相关细节，包括模型原理、线上效果、实践经验及思考。

DSSM 模型

DSSM 模型原理

Deep Semantic Similarity Model 简称 DSSM，是微软于2013年提出的深度学习网络结构，该网络模型将不同结构的信息表示到同一个语义空间中，本质上是实现两种信息实体的语义匹配，基本思想是设置两个映射通路，将两种信息实体映射到同一个隐含空间，在这个隐含空间，两种信息实体可以同时进行表示，便于利用匹配函数进行相似度的刻画。

DSSM 模型最初被应用在检索场景下，通过搜索引擎里海量的点击曝光日志，用 DNN 把 Query 和 Title 表达为低维语义向量，并通过 cosine 距离来计算两个语义向量的距离，最终训练出语义相似度模型。在推荐场景下，一端对应着用户信息，另外一端对应着 Item 信息，DSSM 能够探索用户和物品两种不同的实体在同一个隐含空间内的相似性，进而进行推荐。例如用户会在 App 上检索相关信息，我们可以获得用户点击 Item 的日志，然后通过 DSSM 模型将用户 Query 以及点击的 Item 进行建模，挖掘用户的潜在的偏好，捕捉用户的兴趣，这样便于为用户产生更为精准的推荐结果。

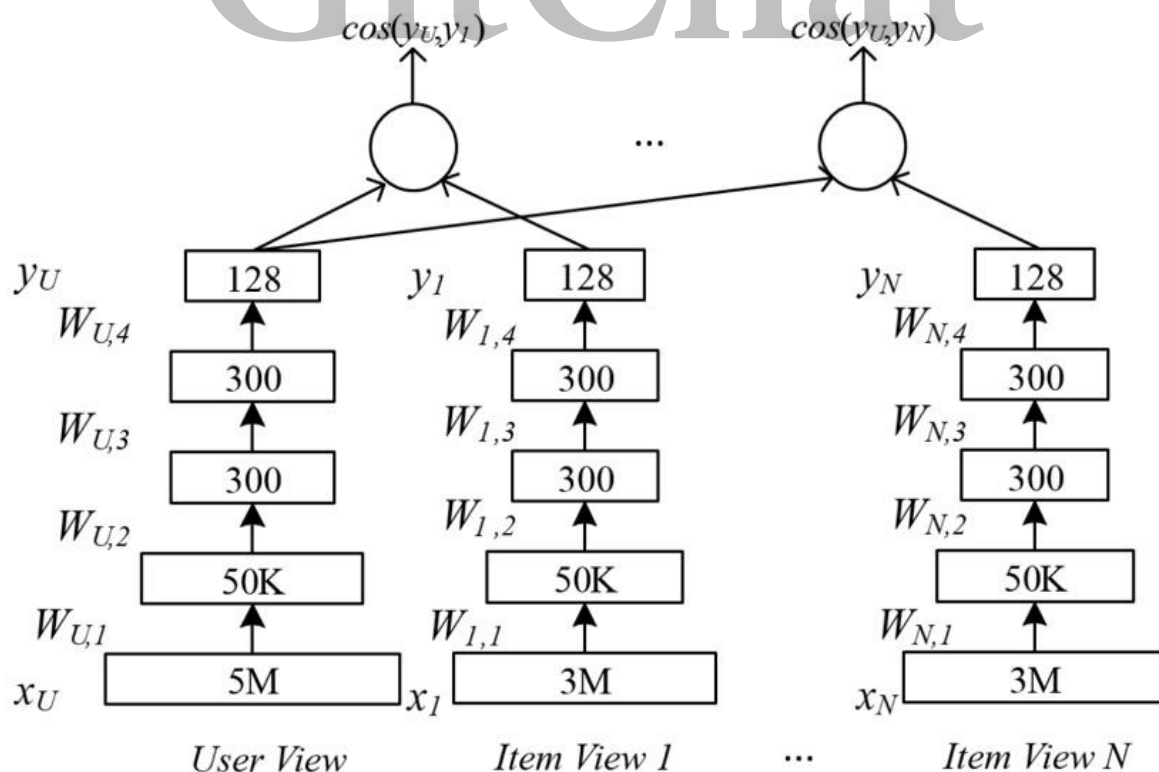


图3 DSSM 模型框架

DSSM 模型训练与预测

(1) 数据清洗与采样

为了增强模型输入的匹配对相关性，提高训练输入数据的 TPR，降低模型随机采样时的 FNR。

1. 行为共现次数：当同一个匹配对多次出现时，说明该对是正例的可能性较大。
2. 同类目过滤：限制匹配对两端内容含有共同类目，强化文本的相关性。
3. 行为时间间隔过滤：匹配对时间间隔相差过长，匹配对两端的相关性弱，需要做过滤。
4. 页面停留时间：过滤用户在内容页停留较短的匹配对，停留时间长短一定程度上表达了用户对匹配对的认可程度。

(2) Query 处理

采用线上词典，对 Query 进行分词得到关键词 list，并进行以下过滤：

1. 过滤特殊符号、拼音及数字。
2. 词性过滤：保留名词、名形词、人名、地名、店铺名、名动词等。
3. 词频过滤：过滤低频词汇，否则会导致训练参数过多，无法收敛。

(3) 文章内容处理

用户在点击文章之前，通常只看见标题部分。但是文章的内容远不止标题，对于文章的 Embedding，我们进行了实体的抽取，在实际测试中，使用 entitykeywords 能得到较好的 validate loss 和推荐效果。

(4) 模型训练与预测

完成以上数据预处理之后，采用微软的 CNTK 进行训练，实验效果如下：

表1：不同数据来源实验数据：

No.	key	value	embedding size	loss-uv
1	内容搜索关键词	搜索列表页点击文章title分词	12194: 9146	1.2735
2	历史搜索关键词	历史点击文章title分词	16742: 11020	2.007
3	历史点击序列关键词	下次点击文章title分词	15352: 15537	2.7623

DSSM 模型效果

(1) 模型线上示例

DSSM 通过将不同类型的文本转换成向量，映射到同一个语义空间中，衡量语义映射的准确性是必要的。但是语义通常存在“一词多义”现象，导致语义关系的定量衡量非常困难，通常需要大量的人工标注，为了排除主观性，有时还需要交叉评测。相关性计算

中，人工评估相关性是必要的，tensorflow 提供了将向量可视化，并可以计算 cosine 或者 euclidian。下图是上海市的部分 Query 和文章内容经过 DSSM 转化成向量后，使用 tensorflow 可视化的效果：

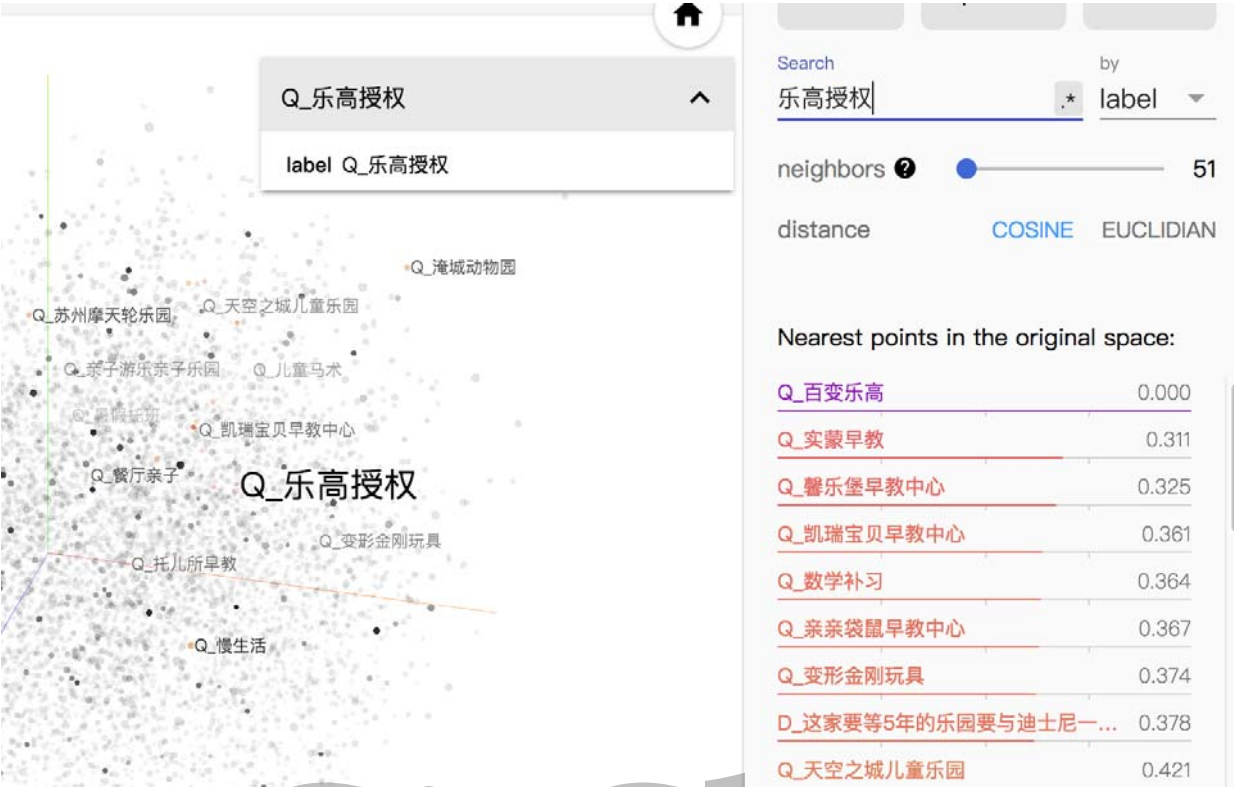


图4 DSSM 模型效果

可以看到模型训练出的语义空间，在图4中对于 Query“乐高授权”，余弦相似度推荐的 Query 或内容在语义上都存在较高的相关性。并且相对文本的内容匹配而言，有更好的语义扩展性能。

(2) 线上效果

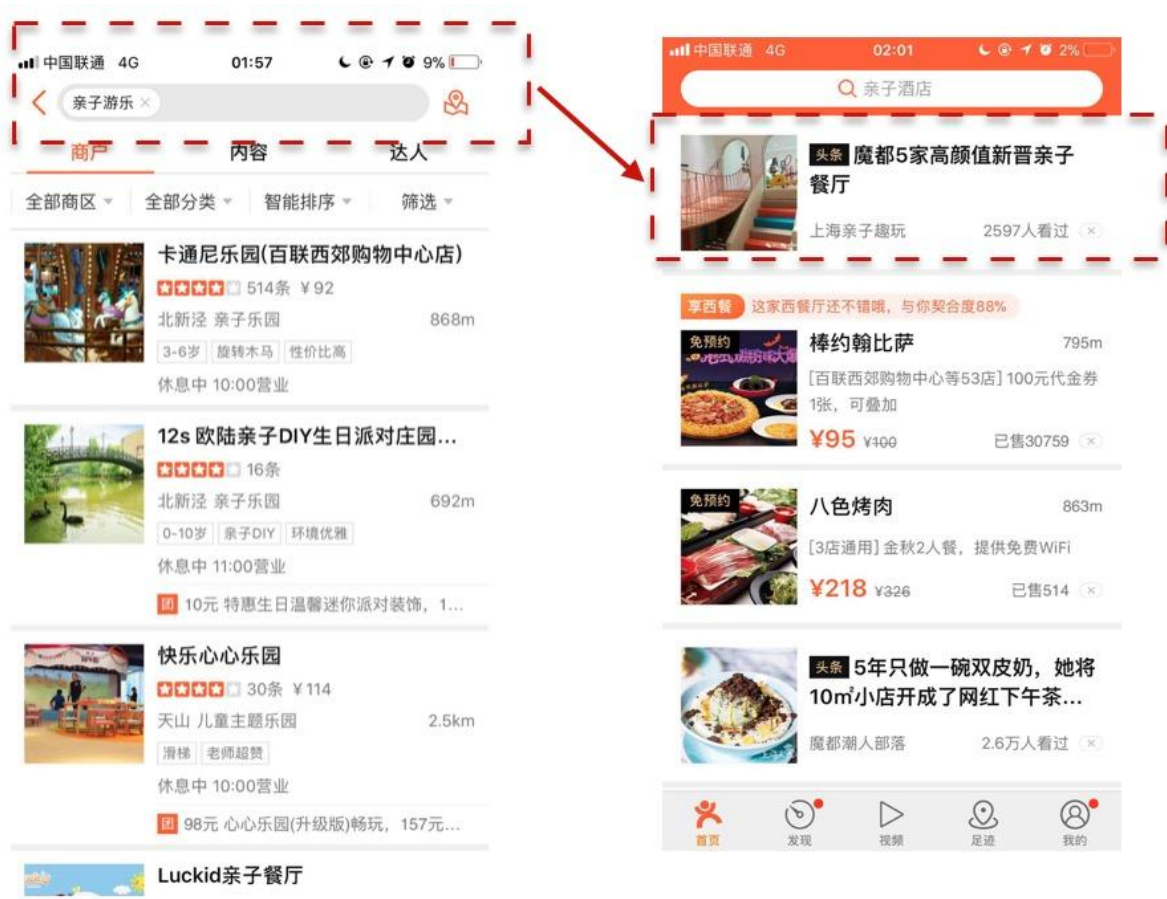


图5 DSSM 模型线上示例

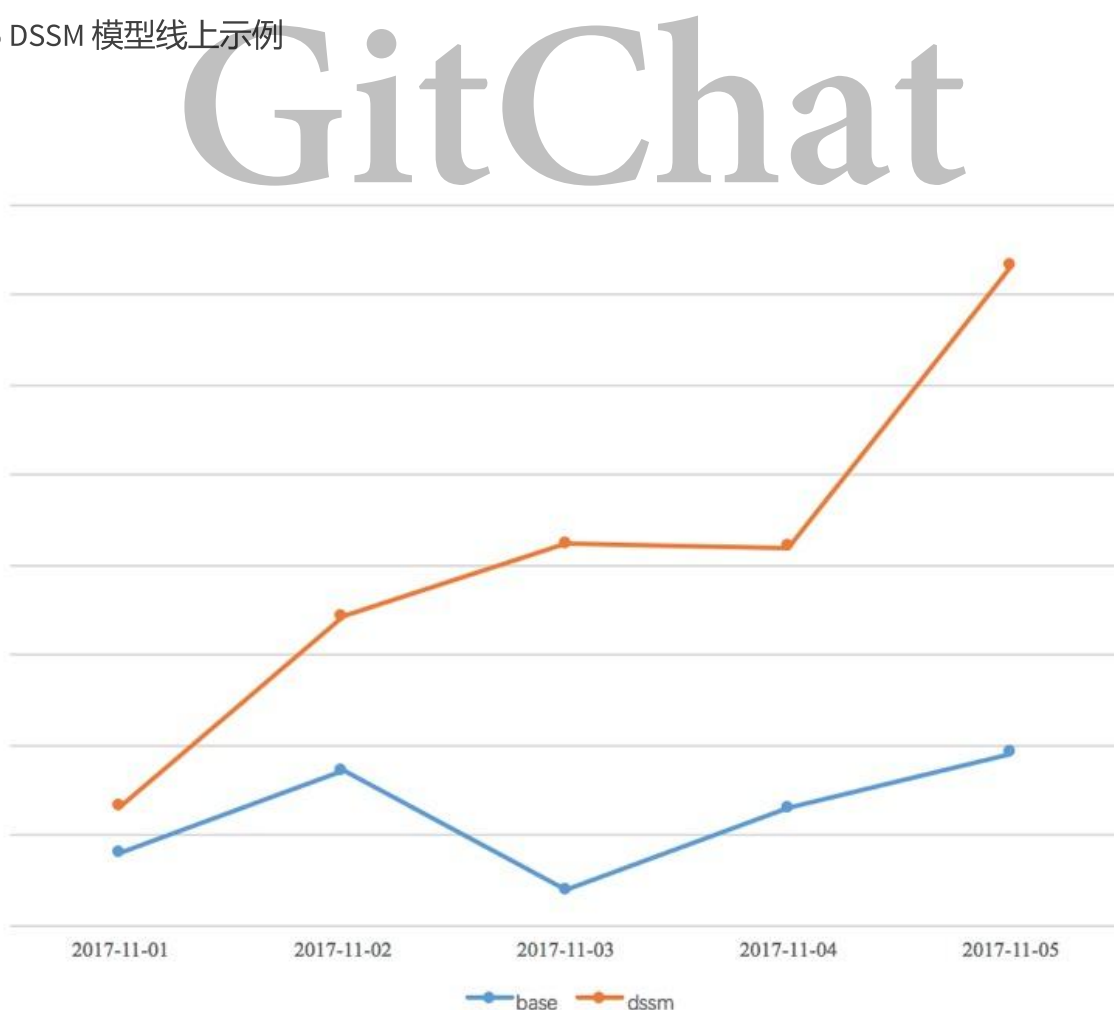


图6 DSSM 模型线上业务提升

在线上应用中，当用户搜索词“亲子餐厅”这样的 Query，右图相关的内容中就会被推荐出来，在实际业务的 AB 测试中，从图6可以看出，DSSM 模型也有比较明显的效果提升。

Session Based RNN 推荐

RNN 序列建模原理

互联网产品的场景下，在用户访问产品页面的一定时间内，其停留的页面、产生的动作会形成一个序列。用户的访问行为是连续的，因此这个序列中的元素具有前后依赖关系。以往的经典推荐算法，比如基于内容的推荐与协同过滤算法，在处理这种序列数据方面存在缺陷：假设 Item 相互独立，不能对同一 Session 中的 Item 连续偏好信息建模。RNN（循环神经网络），是一种特殊的深度学习算法，RNN 可以对前面的信息进行记忆并用于当前输出计算中，通过挖掘序列中的规律，根据用户短期的行为做推荐。深度学习目前已经在推荐系统领域取得一定成果，RNN 应用在推荐的序列建模中，也取得了较好的效果。标准的 RNN 的隐藏层公式如下：

$$\mathbf{h}_t = g(W\mathbf{x}_t + U\mathbf{h}_{t-1})$$

上式中 g 是激活函数，例如可以选择 Logistics 函数，式中 \mathbf{x}_t 是 t 时刻的输入，RNN 会根据前一个状态 \mathbf{h}_{t-1} 与当前状态的输入，计算下一个状态的输出 \mathbf{h}_t 。传统的 RNN 面临着“梯度弥散”的问题，在实际应用中，通常会选用 LSTM、GRU 等新型 RNN 网络。LSTM 能够学习到长期依赖关系，解决传统 RNN 的短板，目前已经成功应用到很多领域。

模型搭建

（1）模型网络结构

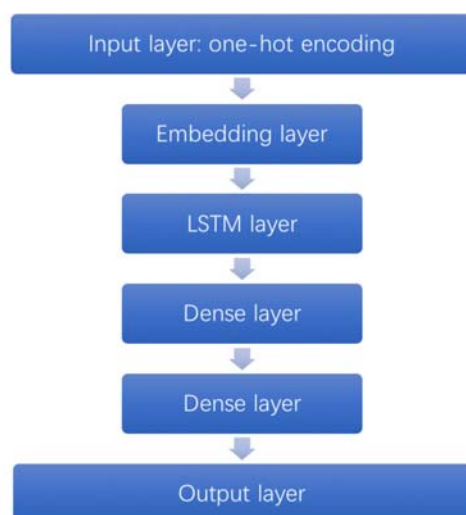


图7 Session Based RNN 推荐网络结构

我们根据用户历史点击的 Item 序列预测用户未来点击的 Item。模型框架如上图所示。首先输入层对用户访问序列中的 Item 进行 one-hot 编码，在输入层之上通过 Embedding 层压缩成稠密连续值，在 Embedding 层之上我们搭建了若干个 RNN 层，最后是增加若干全连接层到输出层。输出层的 loss 目前采用的是交叉熵。

(2) 训练数据选取

训练数据构造我们先后尝试了两种方式。一种方法是，同一 Session 中，依次将前一个 Item 与后一个 Item 分别作为输入与输出进行配对。一个 Session 中可能抽取多对，从不同的 Session 中连续抽取多个输入输出，配对作为 mini batch 进行训练，如下图所示。

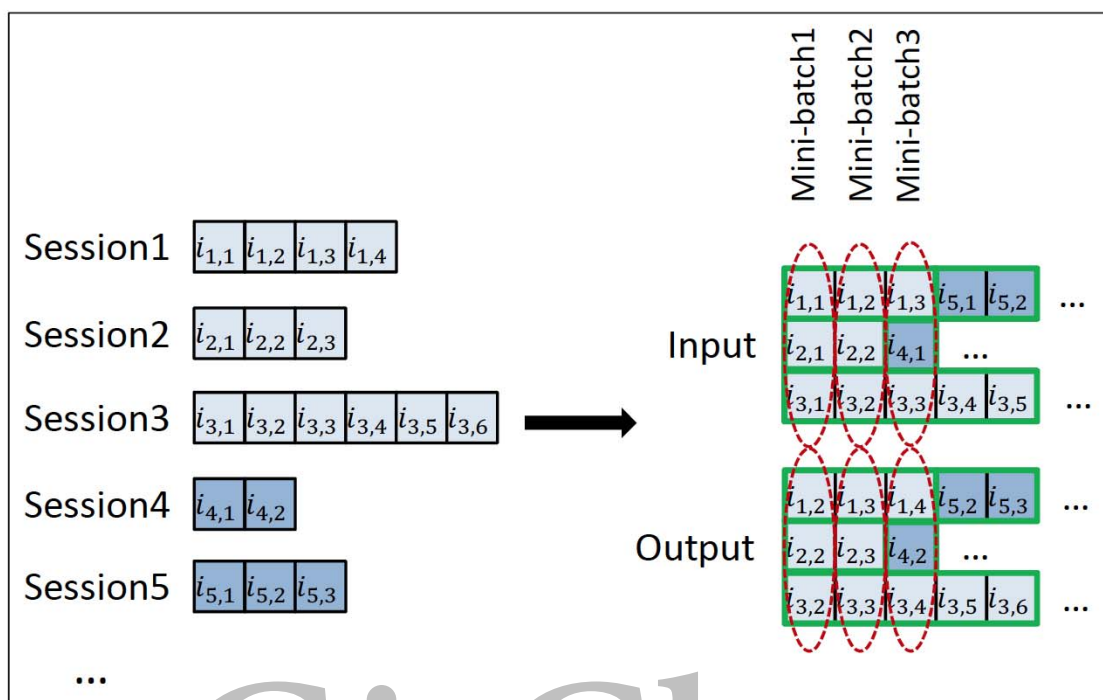


图8 基于 Session 的 mini batch 训练对构造

另一种方法是同一 Session 中前 N 个 Item 作为输入，后一个 Item 作为输出。第一种我们认为是 1-to-1 的预测，与传统的基于 Item 的协同过滤算法类似；第二种是 N-to-1 的预测，能更有利于挖掘序列中 Item 之间的依赖信息。在计算量上，N-to-1 的方式实际预测时只需一次计算，比 1-to-1 的方式节省计算与时间。

(3) 损失函数的选取

在模型的训练过程中，损失函数的选择影响着模型的效果，在我们实际推荐项目中，考虑到具体业务场景，我们采用 Point-wise cross entropy 作为损失函数，采用这个损失函数计算方式比较简单，对正负样本有比较明确的界定，方便计算。

(4) 线下评估

选取点评 App 上的头条文章点击数据为样本，时间区间为1个月，亿级的样本。采用 recall@20、accuracy 两个指标进行离线评估。离线评测阶段，对不同网络结构进行对比，下表对比了使用 Embedding 层与不使用 Embedding 层的离线效果，结果如下：

表2：不同数据来源实验数据：

	accuracy	recall@20
RNN, embedding	0.1311	0.5827
RNN, no embedding	0.0051	0.0642

通过上表2，可以看到采用 Embedding，对于提升模型的效果非常重要，这是因为深度学习是一种表示学习，对稠密，连续的数据比较有效，通过 Embedding 将高维的 one hot 特征，进行层层编码，逐渐压缩到一个比较低维度的空间上，也利于模型的稳定。

(5) 线上效果评估

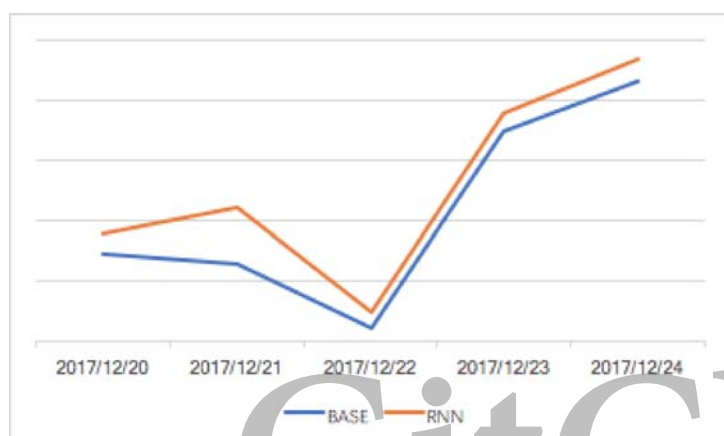


图9 RNN 推荐线上效果评估

表3：RNN推荐线上不同子策略对比

子策略名称	点击率
HotRank Strategy	3.58%
ItemCF Strategy	4.25%
RNN Strategy	4.85%

图9中，RNN 推荐策略在线上的 AB 实验测试效果稳定，业务指标提升明显，可以为用户提供更好的推荐结果，并且我们还可以看到 RNN 子策略的效果也要好于 Item CF 及热门策略。

模型参数优化经验

在线下实验的过程中，对不同的网络结构进行实验，得到如下结论：

- Embedding 层非常重要：高维稀疏 ID 特征一般都需要 Embedding 之后再进入深度学习模型。因为 one-hot 编码不能捕捉相似 ID 之间的相似关系，而通过模型训练得到 Embedding 之后的低维连续向量，能够捕捉相似 ID 的相似关系，并且低维连续向量使得深度学习训练更稳定。
- 增加 dropout：针对用户兴趣变化和误点击的问题，用 dropout 的方式来增强模型泛化能力，提高模型的稳健性。

Wide&Deep 排序模型

推荐系统的整体架构，由两个部分组成，召回系统和排序框架。首先，需要根据用户、物品及场景信息，按照推荐算法模型，产生一批候选结果。完成候选结果之后，需要对候选的物品进行排序。在工业界的推荐系统中，常用的线性模型（例如 logistic regression）被广泛使用，因为它的简单性、可扩展性以及具有较强的可解释性，同时线性模型的性能好，能够满足工业界大规模高并发的性能需求，线性模型通常会使用 one-hot 编码的稀疏特征，具有很好的记忆性。但线性模型比较难挖掘 LR 模型对于高维稀疏的数据有非常好的性能，并且可以通过交叉特征来学习与目标 Label 之间的关联。线性模型缺点是需要人工的去设计交叉特征，同时模型比较简单，无法很好的挖掘用户与候选产品之间的内在的非线性关系。深度模型可以很好的融合物品的信息，挖掘用户和物品之间潜在的关联，但由于它的高复杂度，对于高维的稀疏数据存在比较大的问题。正因如此，Google 提出了 Wide&Deep 排序框架，它是一个联合模型，参数是联合训练的。在 Wide&Deep 模型中，深度模型可以挖掘用户与物品，以及各个特征之间的深层关联关系，而广度模型可以使用更少的参数高效地记忆特征与目标 Label 之间的线性关联。

点评 Wide&Deep 排序框架

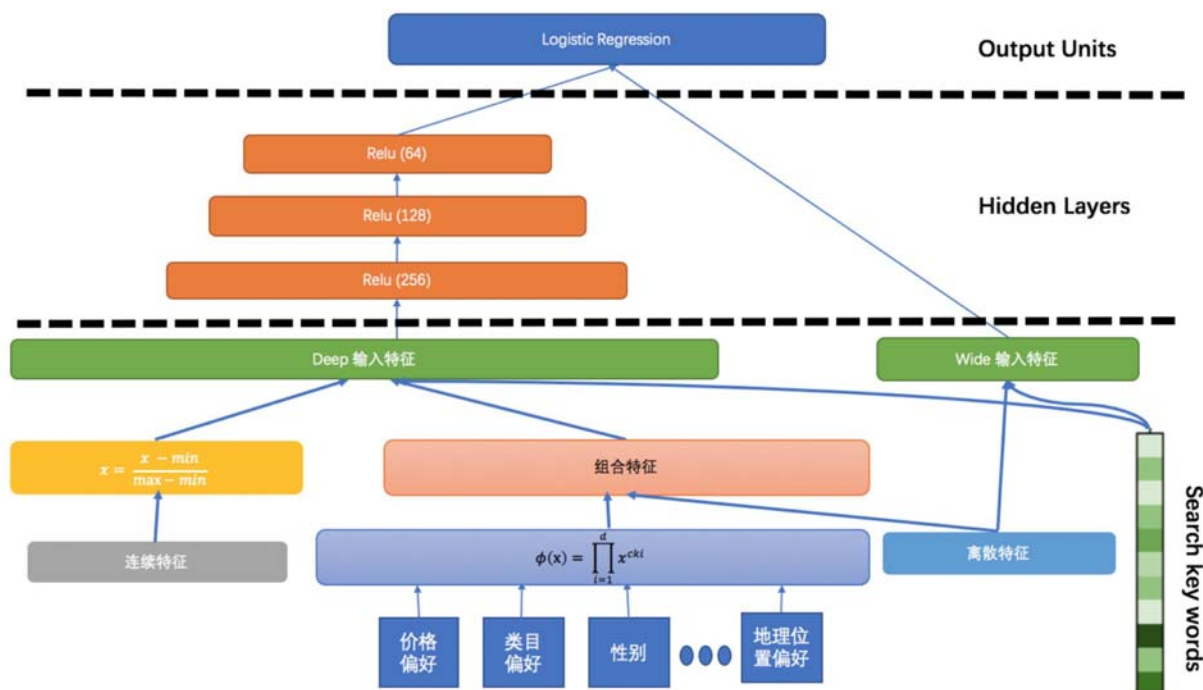


图10 Wide&Deep 排序框架

Wide 组件和 Deep 组件组合在一起，对输入的样本进行预测，它会被 feed 给一个常见的 logistic loss function 来进行联合训练，联合训练会同时优化所有参数，通过将 Wide 组件

和 Deep 组件在训练时进行加权求和的方式进行。我们会将一些连续的特征进行归一化，并将一些低维的离散特征进行 Embedding 输入 Deep 网络，同时对于一些高维离散特征会被输入到模型的 Wide 部分。我们结合业务需求，挖掘出一些比较有意义的组合特征，这些组合特征也可以一定程度的提升模型的效果。

Wide&Deep 模型线下实验

Wide&Deep 模型有着通用性很强的网络架构，有非常多的可调参数，在模型方面我们用 Adam 作为优化器，用 Cross Entropy 作为损失函数。在训练期间，与 Wide&Deep Learning 论文中不同之处在于，我们将组合特征作为输入层分别输入到对应的 Deep 组件和 Wide 组件中，在线下，我们采用基于 Theano、Tensorflow 的 Keras 作为训练工具，训练样本和测试样本均为千万规模，针对不同的网络结构，进行了不同的离线实验，效果如下：

表4：Wide&Deep基础实验数据：

算法模型	AUC
基础模型	0.6885
Deep learning	0.6998
Wide &Deep	0.7065
Wide&Deep (Cross Feature)	0.7181

从表4可以看出，Wide&Deep 模型的离线 AUC 有明显的提升，采用交叉特征可以明显的提升模型的效果。

表5：Wide&Deep网络不同层宽实验数据：

隐层宽度	AUC
512 ReLU	0.7221
256ReLU	0.7181
256ReLU->128ReLU->64ReLU	0.7166

从表5可以看出，Wide&Deep 模型的离线 AUC 随着隐藏的宽度提升，也有相应的效果提升，但考虑到线上的性能，我们对目前使用的层宽未做进一步扩展。

Wide&Deep 模型线上部署及效果

(1) 线上部署

在线上，我们采用 DeepLearning4J 进行部署，DeepLearning4J 不是第一个开源的深度学习项目，但与此前的其他项目相比，DL4J 在编程语言和宗旨两方面都独具特色。DL4J 是

基于 JVM、聚焦行业应用且提供商业支持的分布式深度学习框架，其宗旨是在合理的时间内解决各类涉及大量数据的问题。同时，DL4J 集成了 Hadoop 和 Spark，设计用于运行在分布式 GPU 和 CPU 上的商业环境。

为了解决 Java 缺少强大的科学计算库的问题，DL4J 开发者编写了 ND4J 这个库。ND4J 在 Java 中的角色类似于 numpy 在 Python 中的角色。与 Java 循环相比，使用 ND4J 进行矩阵运算的速度大大提升，主要原因是 ND4J 底层调用了 BLAS（numpy 也是）。

Wide&Deep 模型的海量参数，不仅使得模型离线训练时间大大增加，更严重的问题是在线的预测速度也被拖慢。我们在线使用 DeepLearning4J 进行预测，相对于普通的矩阵计算算法，DeepLearning4J 在线上服务性能提升明显，服务的预测时间压缩到30ms以内，这样未来模型的规模与特征维度都可以做进一步扩展，便于我们尝试更复杂的模型和更丰富的特征。

（2）线上测试效果

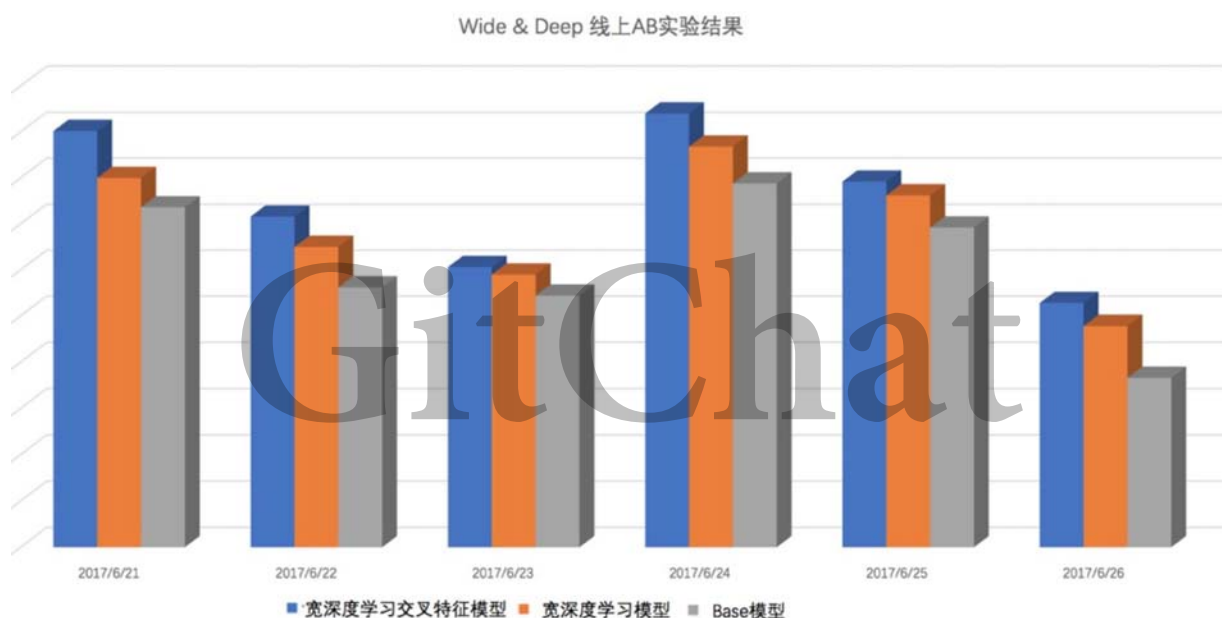


图11 Wide&Deep 排序线上效果

从上图可以看出，Wide&Deep 模型在美团点评推荐上，取得了非常明显的效果提升，并且通过对在线服务部署的优化，线上性能也可以满足要求。

总结与思考

深度学习模型具有非常强的表达能力，并且在图像及语音等领域取得了巨大的进步。在推荐领域，深度学习的相关应用还处于起步阶段，相信未来会有更大的发展和进步。本文在介绍了点评推荐平台业务构建相关背景的基础上，介绍了深度学习模型在点评推荐业务中的应用，包括 DSSM 深度语义推荐模型、Session Based RNN 推荐模型以及 Wide Deep Learning 的排序框架，我们结合实际的业务背景，对这些模型进行了应用，并取得了明显的效果，有效的推动了业务的进步。在未来，我们还需要在原有的工作基础上，做进一步的实践和探索，例如：

- 我们会考虑尝试更为复杂且有业务价值的深度网络，提升模型的特征抽象及学习能力，同时还需要更多的考虑 User 和 Item 之间的一些上下文及内容相关的信息，进一步提升推荐算法的业务效果。
- 使用更复杂的深度网络，除了考虑模型的推荐性能，还需要考虑模型的裁剪及压缩，使得模型可以进行更为高效、更为稳定的进行预测。

考虑端到端的结构，开发适用于不同推荐场景的通用深度学习网络架构，打造更为强大的智能推荐大中台。

参考文献

1. Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013.
2. Elkahky, Ali Mamdouh, Yang Song, and Xiaodong He. "A multi-view deep learning approach for cross domain user modeling in recommendation systems." Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2015.
3. Shen, Yelong, et al. "Learning semantic representations using convolutional neural networks for web search." Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014.
4. H. Cheng, L. Koc, J. Harmsen et al, [Wide & Deep Learning for Recommender Systems](#), DLRS 2016 Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.
5. P. Covington, J. Adams, E. Sargin, [Deep Neural Networks for YouTube Recommendations](#), RecSys '16 Proceedings of the 10th ACM Conference on Recommender Systems.
6. H. Wang, N. Wang, D. Yeung, Collaborative Deep Learning for Recommender Systems, KDD '15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
7. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DSSM_cikm13_talk_v4.pdf
8. Hidasi, Balázs, et al. "Session-based recommendations with recurrent neural networks." arXiv preprint arXiv:1511.06939(2015).
9. <https://keras.io/>
10. <https://github.com/hidasib/GRU4Rec>
11. 深度学习在美团点评推荐平台排序中的运用