**人工智能篇**

[人工智能篇](#page11) [1](#page11)

[机器学习在酒店呼叫中心自动化中的应用](#page12) [2](#page12)

[携程度假智能云客服平台](#page15) [5](#page15)

[机器学习算法在饿了么供需平衡系统中的应用](#page21) [11](#page21)

[携程个性化推荐算法实践](#page28) [18](#page28)

[平安银行算法实践](#page37) [27](#page37)

[携程“小诗机”背后的机器学习和自然语言处理技术](#page46) [36](#page46)

[携程图像智能化建设之路](#page55) [45](#page55)

[面向前端工程师的机器学习引导课](#page63) [53](#page63)

[证件全文本 OCR 技术，了解一下](#page76) [66](#page76)

[携程 AI 模型引擎设计与实践](#page91) [81](#page91)

[“猜你所想，答你所问”，携程智能客服算法实践](#page98) [88](#page98)

[知识图谱在旅游领域有哪些应用？携程度假团队这样回答](#page106) [96](#page106)

[如何选出最“美”图片展示给你？携程做了基于深度学习的图像美感评分系统](#page111) [101](#page111)

[全球顶级算法赛事 Top5 选手，跟你聊聊推荐系统领域的“战斗机”](#page117) [107](#page117)

[携程实时智能异常检测平台的算法及工程实现](#page123) [113](#page123)

[行业智能客服构建探索](#page131) [121](#page131)

人工智能篇



**机器学习在酒店呼叫中心自动化中的应用**

**[作者简介]**周振伟，携程数据智能部数据科学工程师，同济大学硕士，主要承担酒店服务领域的数据分析和挖掘工作。

无论是出门旅游还是商务出行，在外能有一个舒适的住处，往往都是首先要解决的问题。OTA提供的酒店预订功能无疑为此提供了巨大的便利。

打开携程 APP，看中一家不错的酒店下单后，会有一个等待酒店确认的过程。携程将用户预订的消息发送给酒店，酒店进行查房，确认是否有空余房间，然后回复携程，再由携程通知用户确认结果。这个过程通常在半小时到一小时内完成，很多时候只需十分钟，在这背后，携程的呼叫中心起到了重要的作用。

面对每天全国海量的出行和预订需求，避免用户长时间的等待，加上与酒店相关的各种复杂事宜，呼叫中心的工作一直在高度紧张和繁忙的氛围中进行。然而随着业务量的不断增长，完全靠人力完成这些工作效率是非常低的。在大数据和人工智能时代，我们自然会想到用数据的力量代替一部分人力工作，即实现和提升流程的自动化，提高工作效率。

本文以订单确认智能外呼为例，分享携程在提升呼叫中心自动化方面所做的工作，介绍携程是如何利用机器学习技术，改造呼叫中心外呼流程的。

**一、呼叫中心职能**

携程呼叫中心的职能主要包含以下四部分：

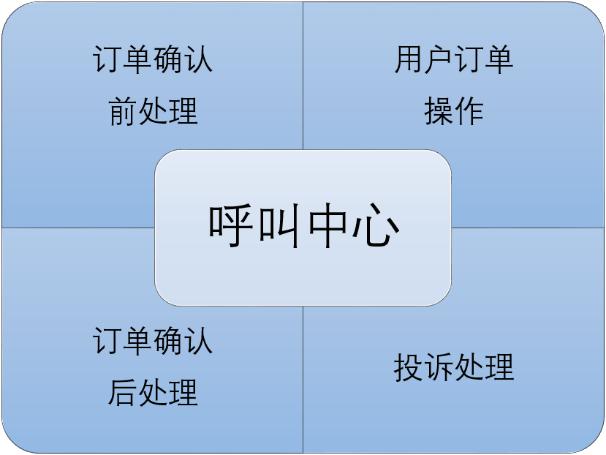
（1）订单确认前处理：从用户下单到确认订单为止的过程，称为前处理。确认订单是指用户下单后酒店确认是否可以正常入住，然后携程告知用户确认结果。除一部分房型可以直接确认用户不用等酒店回复以外，大部分房型都需要由酒店亲自确认。这部分房型只要有用户下单就必然经历这个过程，因此前处理是携程呼叫中心工作量最大的事务之一。

（2）订单确认后处理：从确认订单到用户入住和离店为止的过程，称为后处理。对用户来说，订单确认后只需到那一天到达酒店入住即可。对携程来说还有很多事宜需要和酒店沟通，例如入住人或时间变更、离店后的审核结算等。此外，确认订单时如果酒店告知房间已满无法入住，携程会通知用户并推荐用户改订其他酒店，该过程也属于后处理。

（3）用户订单操作：即通常所说的客服，接听用户来电，处理用户取消订单、修改入住人或时间、增加特殊要求、开发票等方面的需求。

2

人工智能篇



（4）投诉处理：负责处理用户投诉

除处理投诉必须人工协调以外，其他三项职能的工作都有既定的流程，因此都包含大量重复性的工作。经过长期的工作积累发现，这些流程整体保持高效，但部分环节效率略低，可以用自动化的方式代替或改进。

**二、订单确认智能外呼**

上文提到，如果酒店长时间没有确认订单，呼叫中心会人工外呼给酒店催单。问题是，应该采取怎样的外呼策略，可以使订单确认外呼这项工作更加高效。

**2.1 传统的订单外呼流程**

传统的订单确认流程很简单，等待酒店回复——到指定时间仍未回复——人工外呼酒店。这是一个很自然的流程，然而实际效率并不高。主要有两点：

（1）向酒店发出预订消息后，只能一直等待，直到快要超时了再去催单吗？事实上，有些酒店确认订单的速度一向很慢，或者某个特殊的时段、某些特定的房型酒店总是很晚才给回复。可能我们已经预料到酒店会很晚才给回复，但也会等到指定时间外呼催单，这样用户就白白等待了很长时间。或者，我们可以提前外呼这些酒店催单？

（2）实际操作中发现，很多时候，酒店只是在电话里表示了解情况，通话结束后仍然过很久才给回复，那这次外呼就变得没有意义，可能酒店已经习惯或者确实需要那么多时间来做确认，有没有给酒店外呼结果是一样的。毕竟人力有限，不可能所有订单都做外呼，应该优先去做外呼有意义的订单，而可能无效的外呼就应暂缓。那么，在外呼之前，能否预判这次外呼是否有效？

顺着以上两个问题的思路，就把一个依次排队的傻瓜式外呼，转变为智能化的预测式外呼。这里的关键，就是要提前找出哪些订单需要提前外呼，以及预判外呼的有效性。这是一个需要自动执行的预测功能，机器学习在这里派上了用场。

**2.2 改造后的智能外呼流程**

3

人工智能篇



改造后的预测式外呼流程加入了两个机器学习模型：回复时长预测模型和外呼有效性模型。

（1）通过回复时长预测模型，预测订单的回复时长是否会超过一定容忍范围。在容忍范围内回复的，暂时不用做什么。如果等到容忍范围后仍未回复，再进入队列准备外呼。预测回复时长超过容忍范围的，提前外呼，提前对可能需要很长时间回复的酒店进行催单。

（2）准备外呼前，每个订单由外呼有效性模型做一次判定，即本次外呼是否有效。判定为有效的，优先安排人工外呼；判定为无效的，说明此时电话打出去也没有意义，人工外呼延后。但为了应对模型误判，防止确认超时，我们会用 IVR 自动语音外呼做一次催单。如果 IVR 没有起作用，在剩下的时间里，就对这些仍未确认的订单安排人工外呼。

总结一下这个预测式外呼流程，我们把可能会很晚回复，并且立即外呼也是有效的订单，尽可能优先安排人工外呼；而其他需要较长时间回复，或者打电话不起作用的外呼延后。比起原本订单一个个排队外呼的方式，新流程对资源的分配更为合理。

**2.3 模型的构建**

下面介绍一下上面所用的两个模型。这是两个有监督模型，我们结合业务背景提取相关特征，构建数据集，离线进行模型训练。

以外呼有效性模型为例，决定外呼是否有效的因素有很多，包括下单所处的时间点，几天后入住，是否是特殊或热卖的房型，携程与酒店的合作关系，酒店历史的外呼数量，无效外呼数量，以及酒店的操作习惯等。训练过程中我们尝试了多种模型，最终选择 XGBoost 模型上线。

模型以及新流程上线后，对比上线前后一段时间的数据，总订单量增长了 25%，而实际进行人工外呼额订单占比减少了 1/3，同时确认用户的平均时长没有显著变化，说明在没有影响客户体验的前提下，通过以模型预测进行资源重新分配的方式，减少了大量无效外呼，流程得到了优化，呼叫中心工作的运行效率得到大幅提升。

**三、总结和展望**

以上我们介绍了大数据和机器学习在提升携程呼叫中心自动化方面的一些应用，重点介绍了预测式智能外呼的细节。这些项目的成功上线，验证大数据和机器学习在提升自动化，优化资源分配，改进流程提高效率方面可以发挥重要作用。我们分析现状，寻找流程中的不足和改进点，用机器学习加以改进，在其他项目中也值得借鉴。

未来携程将致力于打造一个全新的、更加数据化智能化的呼叫中心，从而更好地服务用户，为行业树立一个新的标杆。

4

人工智能篇



**携程度假智能云客服平台**

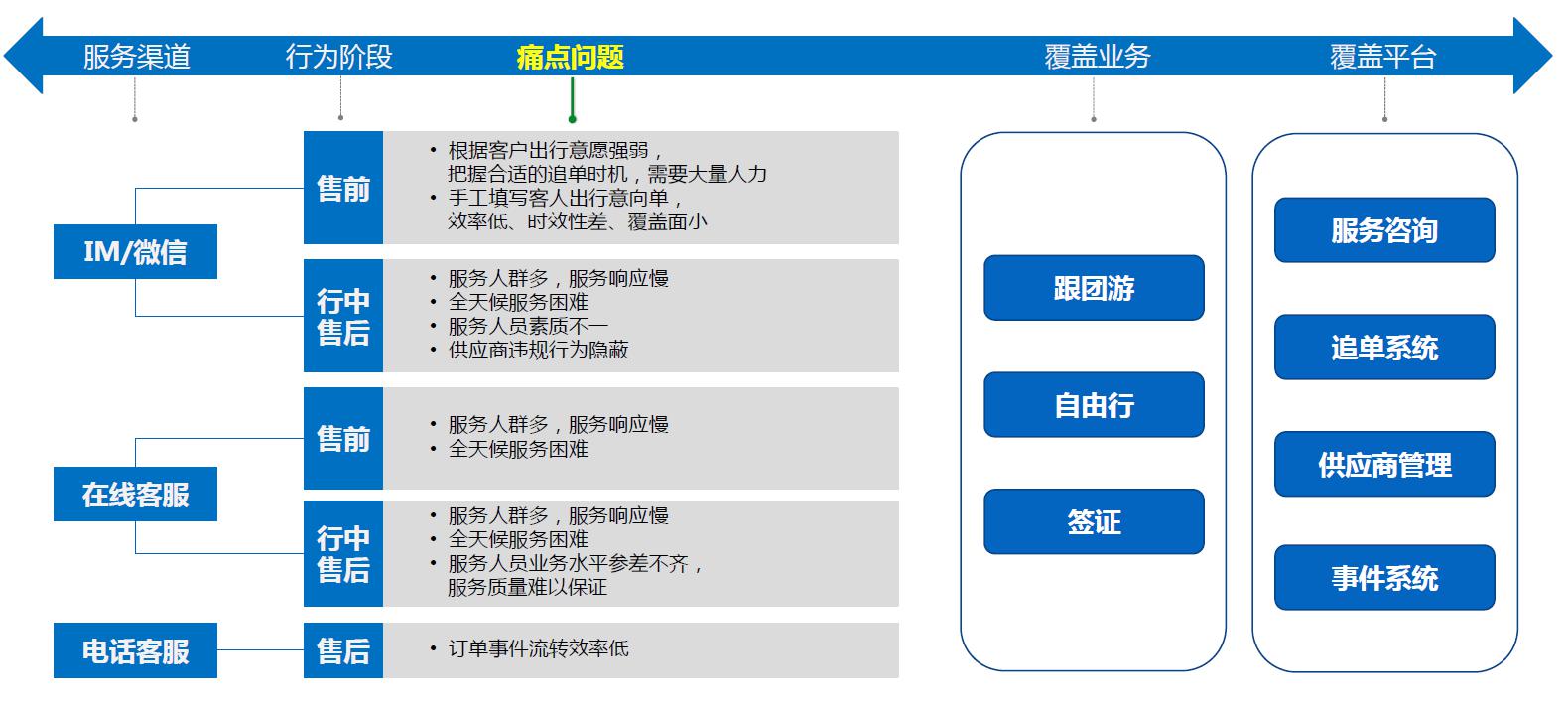
**[作者简介]**李健，携程度假大数据开发总监。2013年底加入携程，在攻略社区及度假负责自然语言处理、图像、推荐等领域的开发管理工作。

**写在前面**

在人工智能时代， AI 技术会以提供更精准更高效的方式在流程改进、沟通费力度下降、沟通效率提高、成本降低及收益提升等众多方面全面改变目前的商业模式、推动业务发展。携程度假的智能云客服平台在这方面做了很多有益的尝试，大大提升了携程度假客服的效率和用户体验。

**一、智能云客服平台概述**

在智能云客服平台上线前，在包括 IM/微信、在线客服和电话客服在内的多个服务渠道的各个行为阶段都存在着不少痛点问题。由于度假的产品涵盖酒店、机票和门票等多个方面，服务的多样性更加明显，给痛点问题的解决带来了更多的困难。



为了解决上述痛点问题，我们建立的智能云客服平台目标主要围绕以下六个方面进行改善：

1）服务流转效率；

2）服务响应时间；

3）标准化服务；

4）自动化服务；

5）违规管理；

6）流程优化；

当前的智能云客服平台已经在智能问答、服务渠道管理、服务流程优化和供应商管理四个方

5

人工智能篇

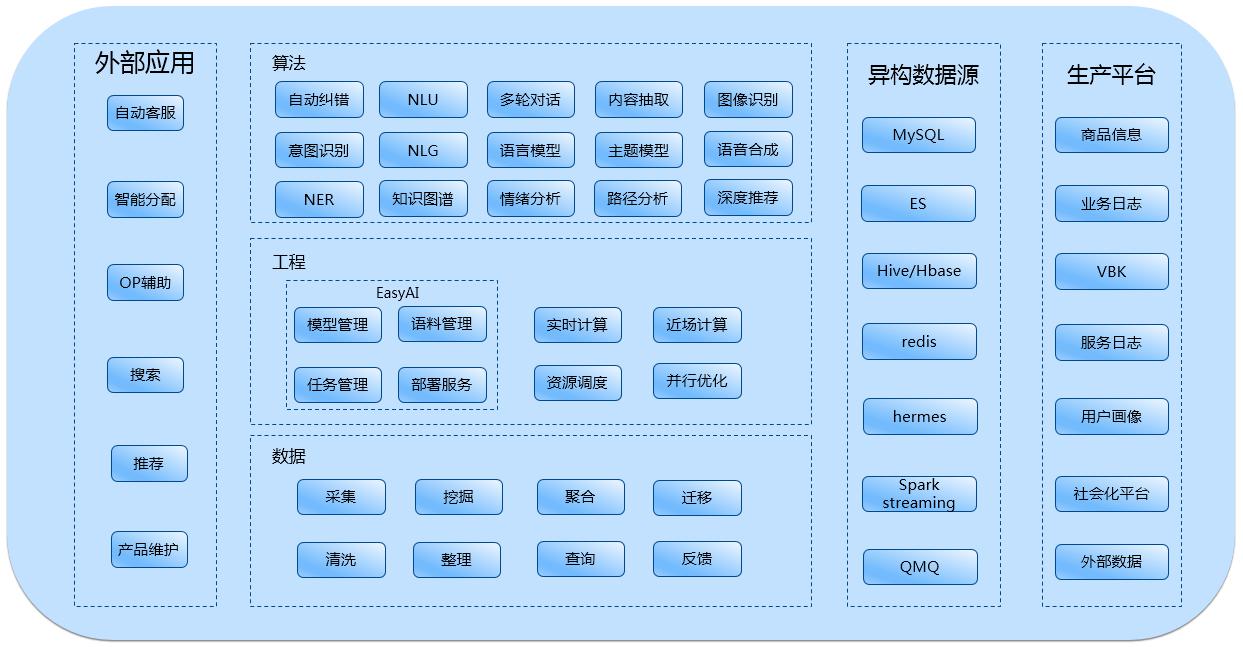


面上线了近 80 个模型，取得了很好的效果。



**二、系统架构**

下图展示了我们智能云客服平台的系统架构，其中在算法部分，我们设计并上线的自动纠错模型，使用户意图理解的准确率从 60%提升到了 90%以上；在工程部分，我们设计并构建了 EasyAI 平台，与很多 AI 平台往往提供给算法工程师使用不同，我们的 EasyAI 面向的是不熟悉技术的业务群体，通过这个平台，提升了业务 50%的工作效率。



**三、典型算法模型介绍**

**3.1 用户意图模型**

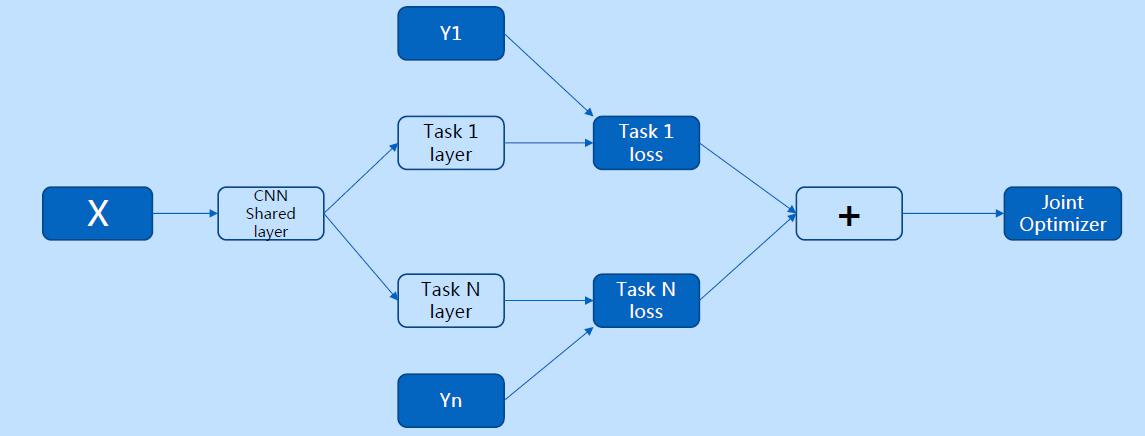
用户在和客服聊天的时候，往往会有各种各样的意图，我们需要实时判断用户是否存在购买某一个具体产品的意图，或者是否能够形成一个明确的意图方便后续的产品推荐。我们采用了基于深度学习的多任务多标签的模型来实现，如果用户被判为存在上述意图，则会基于该

6

人工智能篇

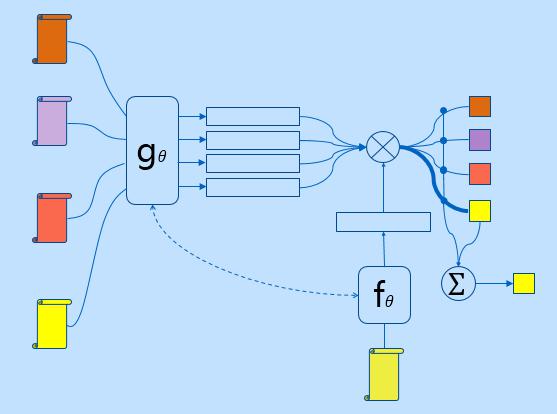


意图，为用户提供更个性化的服务。



**3.2 群监督模型**

因为用户和客服的沟通都是基于群的方式来实现的，所以我们需要使用群监督模型来更好得对用户意图进行分析和判断。在真实情况下，用户的大部分聊天内容比较集中，样本量也非常大，而在某些意图上的样本量缺极度缺乏，为此我们构建了一个有效的群监督模型来解决样本极度不平衡的问题，该模型上线后，准确率从 10%+提升到了 80%+。

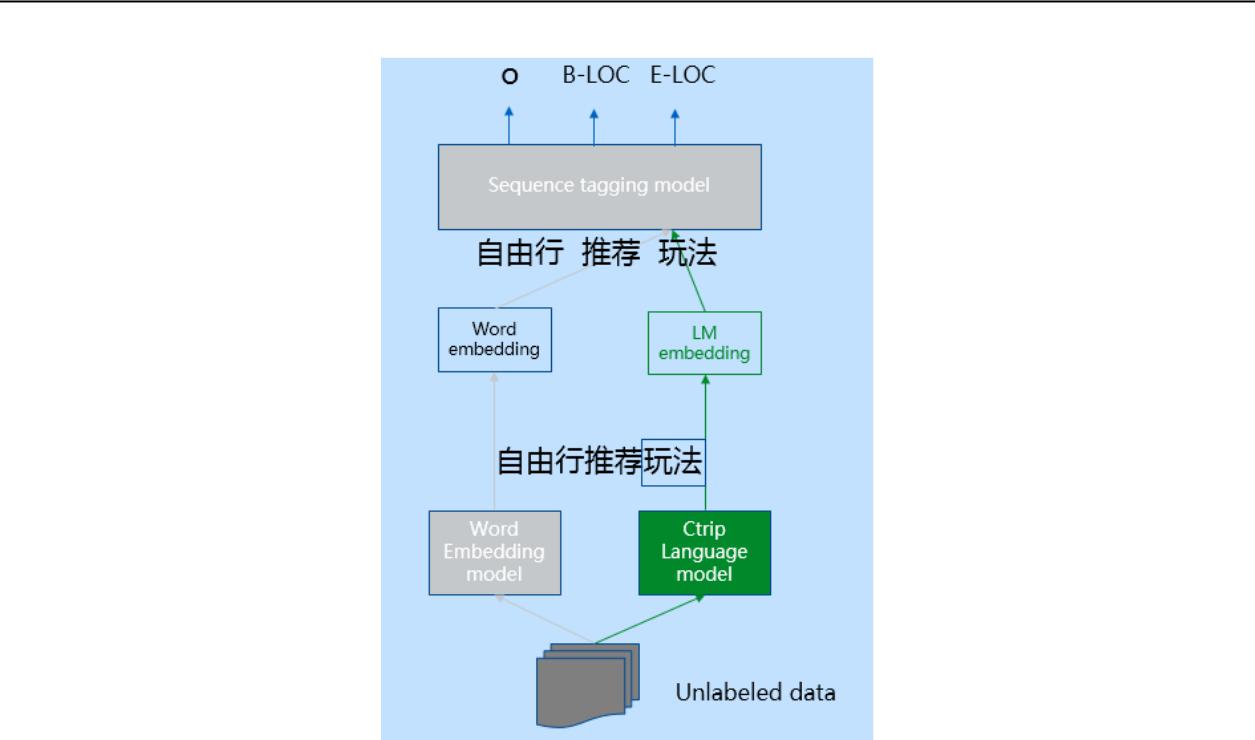


**3.3 内容抽取模型**

在智能云客服平台的诸多应用场景中都会使用到内容抽取功能，为此，我们构建了一个基于半监督学习的内容抽取通用模型。具体地，我们在常用的内容抽取模型的基础上加入自己建立的语言模型来实现半监督学习，最终准确率提升了 7%。

7

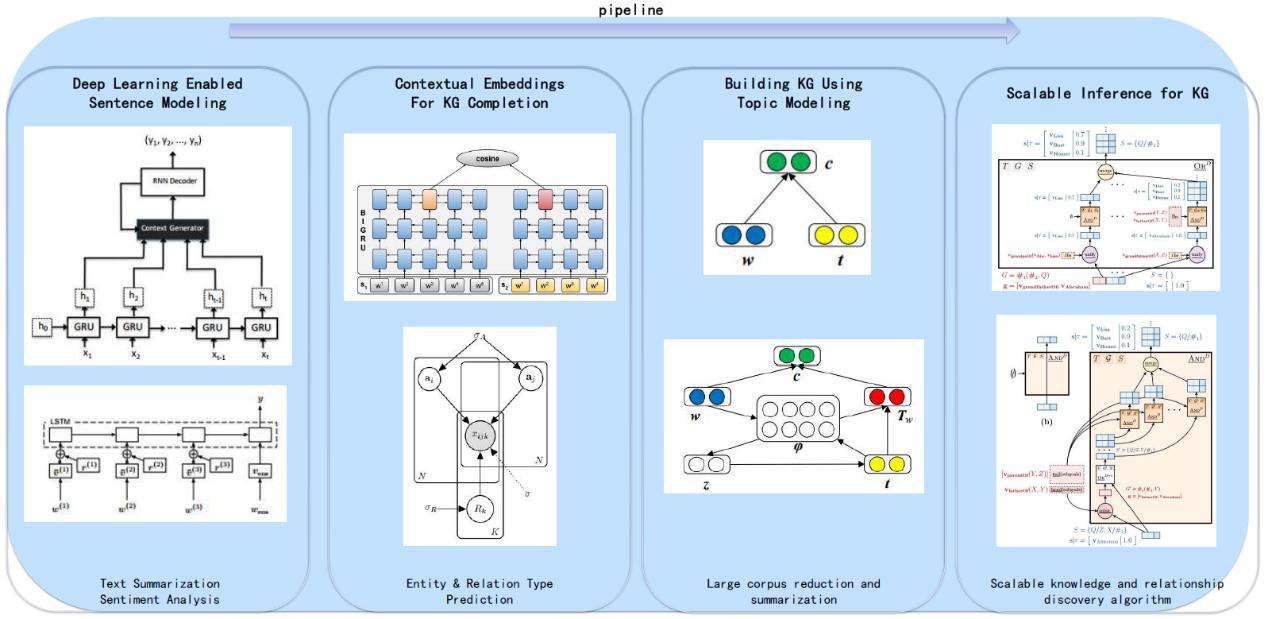
人工智能篇



**3.4 知识图谱构建**

在知识图谱的构建过程中，我们首先划分 Domain，然后在每个 Domain 下去产生一个Schema。产生 Schema 的方式分为人工和自动两种，自动产生 Schema 的效果往往不是很理想，所以我们采用人工定义 Schema 的方式。

接下来我们在 Schema 的基础上实现信息的抽取，并进一步完成知识图谱的补全和融合。下图展示了我们从知识图谱的构建、抽取、补全、融合到推理过程中使用的所有模型。



**3.5 自助问答模型**

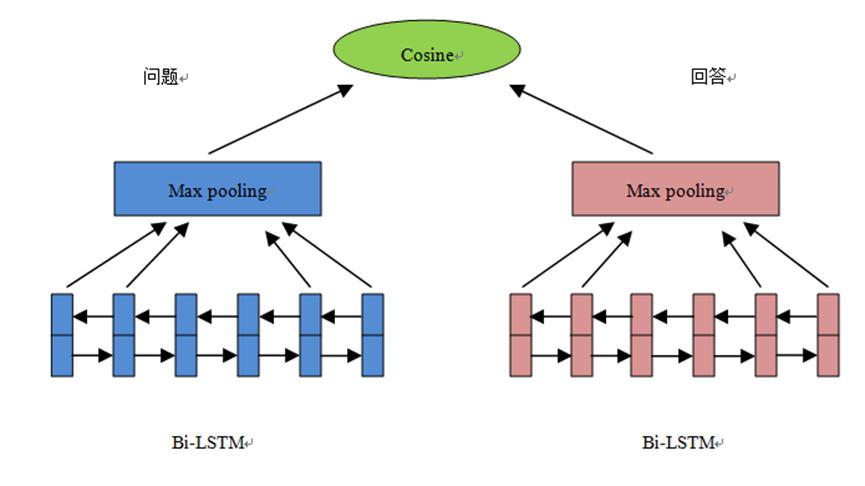
智能云客服平台的自助问答系统经历了从 1.0 到 2.0 的发展过程，在自助问答 1.0 系统中，我们建立粗粒度检索式模型来实现和用户的单轮对话。具体地，我们采用 Bi-LSTM+Attention+CNN 的方式来实现，分别对问题和回答建立模型，最后计算两个模型输出

8

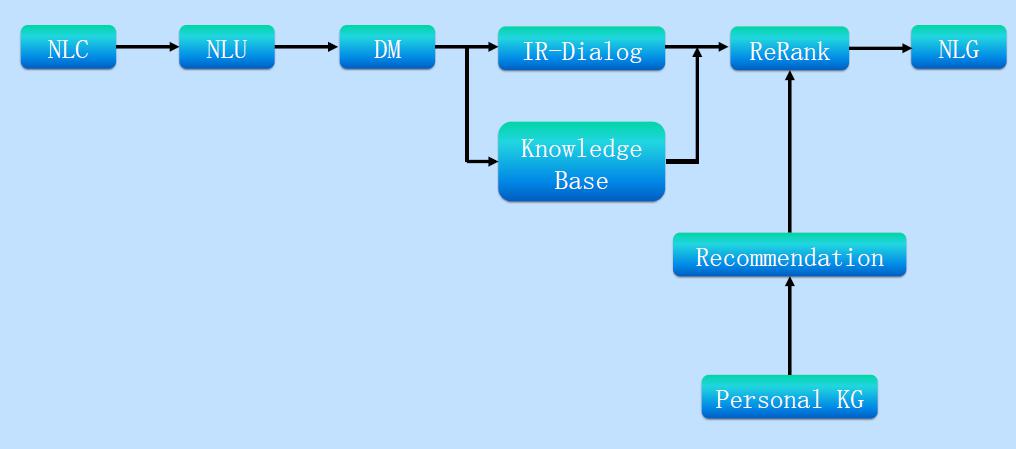
人工智能篇



向量之间的相似度。



在自助问答 2.0 系统中，我们实现了机器和用户的多轮对话。对于用户的聊天内容，首先进行内容自动纠错和自然语言理解，然后进入对话管理，为了保证对话管理的性能，我们开发了核心组件。在对话管理之后，根据对用户意图和对话控制的理解，会选择检索式粗粒度回答和知识库细粒度回答两种方式。在获取所有的回答之后，我们会对其进行重新排序，同时，我们还会根据对用户意图的理解和当前用户所处的环节进行相应的推荐，最后生成最佳回答。下图展示了自助问答 2.0 系统的处理流程。

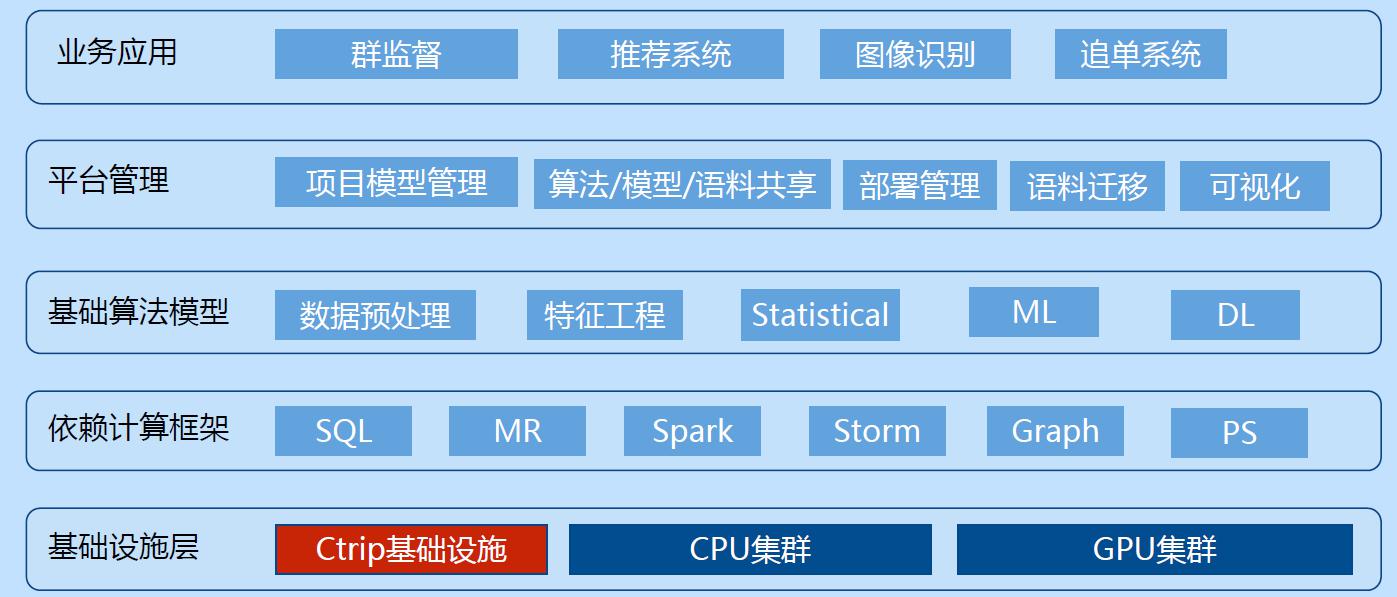


**四、EasyAI 平台介绍**

在第二部分中，我们提到了 EasyAI 平台，这里我们做进一步的介绍。我们建立 EasyAI 平台的初衷是为了直接给业务进行操作。利用 EasyAI 平台，我们可以提高语料标注和模型训练的效率，实现标注语料的共享，同时也可以缩短建设周期，减少模型的重复建设。下图展示了我们 EasyAI 平台的系统架构。

9

人工智能篇



**写在最后**

以上对携程度假智能云客服平台进行了介绍，但 AI 对于携程度假的价值远远不限于此，后续我们会进一步将知识图谱应用于智能推荐和搜索，同时也会实践旅游领域的机器阅读，减少知识图谱构建的费力度，最终实现服务全流程的自动化和智能化。

10

人工智能篇



**机器学习算法在饿了么供需平衡系统中的应用**

**[作者简介]**陈宁，饿了么人工智能与策略部高级算法专家，负责供需平衡系统的算法与研发工作。获新加坡南洋理工大学计算机博士学位，研究方向包括：数据挖掘，机器学习，自然语言处理，软件工程等。本文来自陈宁在第二届携程云海机器学习沙龙上的分享。

即时配送物流系统是外卖领域核心价值所在。区别于其他物流，这个行业的“物流”是希望

1. 分钟内能够送到，这就对我们的算法模型具有很大的挑战。智能调度系统更是即时配送物流系统中的核心。

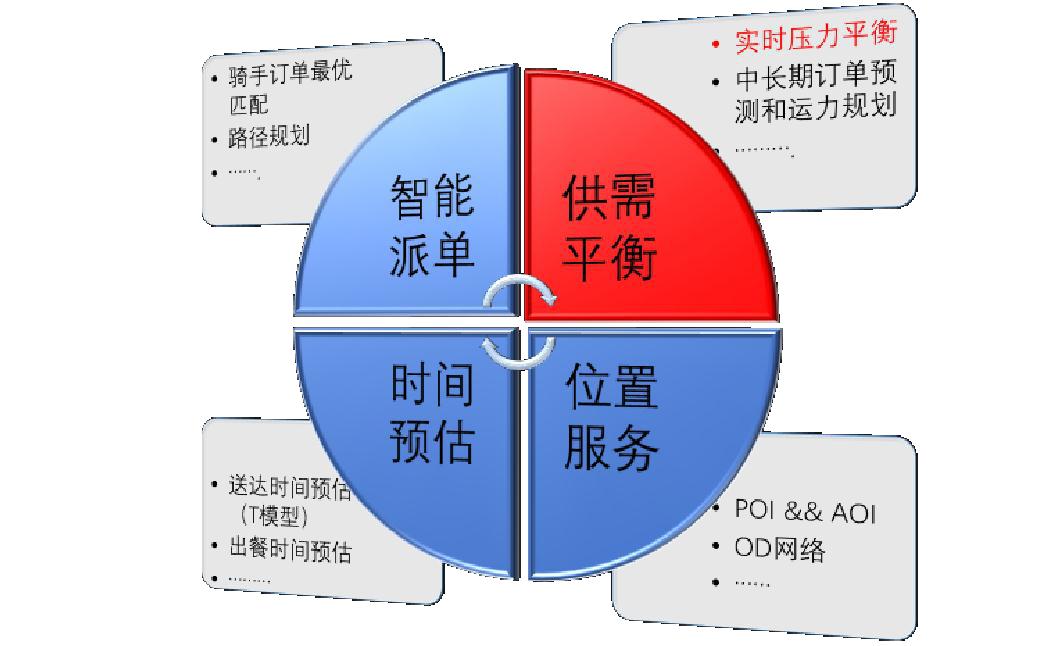
在本文中，我将介绍饿了么的智能调度系统，着重介绍其中的压力平衡子系统，并通过两个实例，分享机器学习算法在构建压力平衡系统中起到的关键作用以及取得的成果。

希望通过本文，帮助一线的机器学习算法工程师和爱好者们了解饿了么即时配送系统中压力平衡系统的构建，以及如何利用常见机器学习算法有效地解决 O2O 场景下的实际问题。

**一、饿了么智能调度系统**

饿了么智能调度系统是外卖即时配送领域中最核心的环节，该系统替代了调度员大部分的工作，减少了人力介入的程度，实现了自动化、智能化的派单。

智能调度系统主要可以划分为四个子系统，如图 1 所示：



11

人工智能篇



图 1 饿了么智能调度系统

* 智能派单：包括骑手与订单的匹配策略, 骑手的路径规划等；
* 时间预估：包括骑手行程时间预估, 骑手楼内停留时间预估, 出餐时间预估与下单送达时间预估等；
* 供需平衡：包括实时的压力平衡，中长期订单预测、运力规划和短期的骑手排班等；
* 位置服务：包括配送范围划定, 商圈/楼宇/配送点的建设，位置校准等。

以上各个子系统相辅相成，组成了整个饿了么智能调度系统。本文中我将重点介绍机器学习算法在供需平衡中的实时压力平衡系统中起到的作用。

**二、压力平衡系统**

**2.1 系统目标**

压力平衡系统要解决的问题是：当配送供给(骑手)与用户需求（订单）出现日内的异常不匹配时，为了保证用户体验不受到过大损伤，及时有效地使用调控手段来平衡需求与供给。

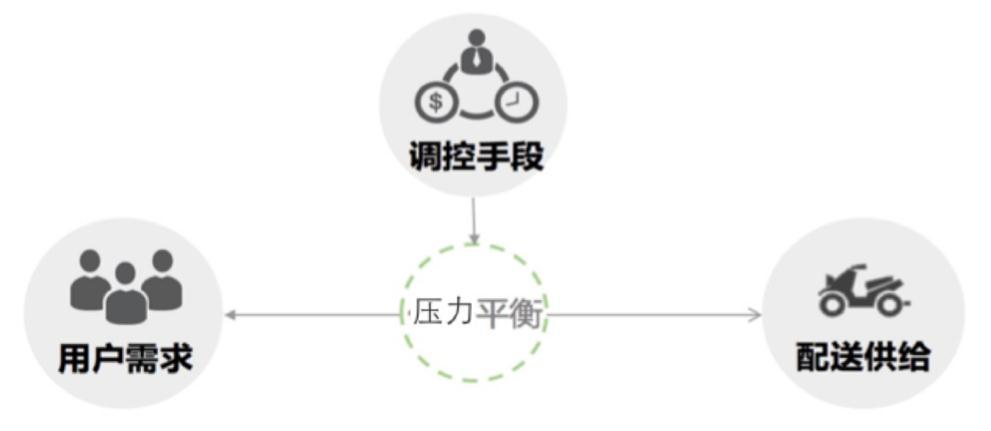


图 2 压力平衡系统目标

造成用户需求和配送供给日内异常不匹配的原因有很多：比如遇到恶劣的天气、商家搞一些临时性的大活动，骑手运力本身不充足等。

当出现异常不匹配时(压力高于某个阈值)，压力平衡系统会采取一定的措施。比如上调配送费、缩小配送范围、下满减活动、关店等。

**2.2 算法框架**

图 3 展示了压力平衡算法的总体架构：

12

人工智能篇

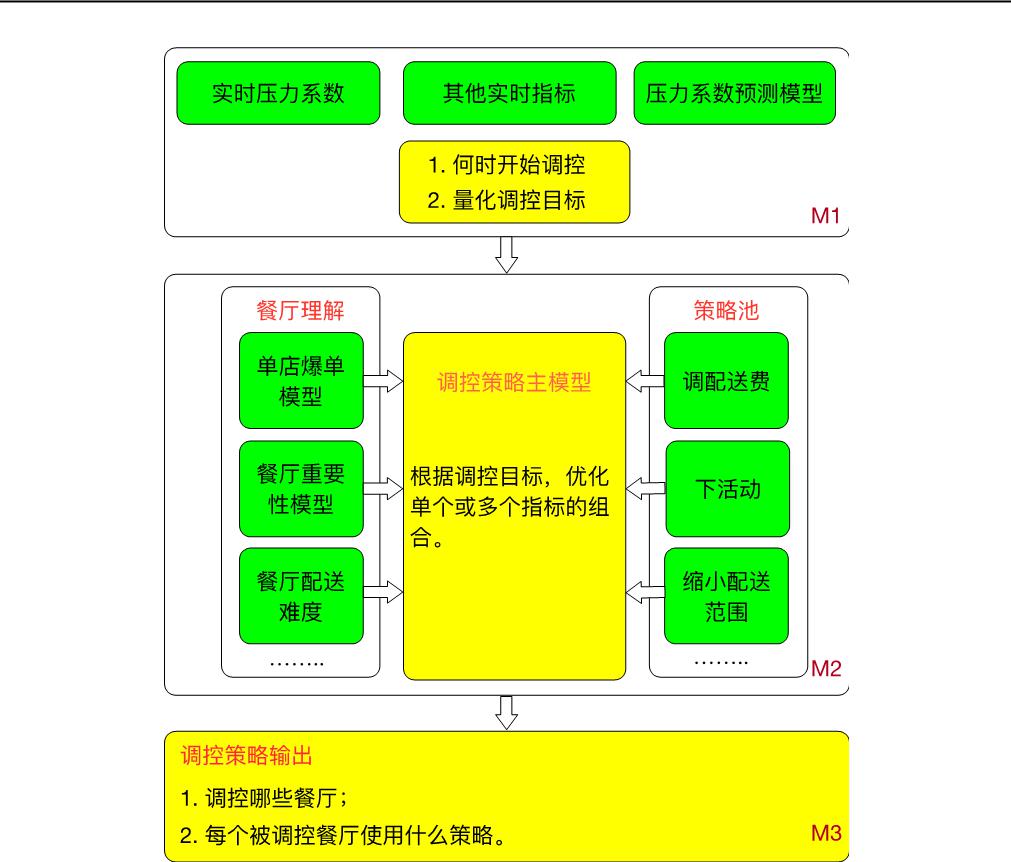


图 3 压力平衡算法框架图

首先，图 3 的最上方 (M1) 展示的是团队(饿了么组织运力的最小单元，一个团队由若干骑手组成) 压力系数及其辅助指标的实时监控和预测模块。这个模块主要的作用是估算自动调控开始的最佳时机和调控的量化控目标：即当压力系数大于一定的阈值时，开始自动采取一系列的措施，使得压力系数降低到可接受的范围。

其次，图 3 的中部 (M2) 展示的主要是两个模块：1）餐厅理解；2）调控策略池。其中，餐厅理解：利用餐厅重要性排序模型，单店爆单模型，餐厅配送难度等信息来细化调控的基本单元，它使我们知道哪些餐厅是应该首先被调控的。调控策略池包含了调控可用的调控手段，例如：增加配送费，下大额满减活动，缩小餐厅的配送范围等。

最后，图 3 的底部 (M3) 展示的是自动调控策略生成主模型：我们会根据量化的调控目标，来优化单个或多个指标的组合。通过数据、算法来决定调控哪些餐厅，每个餐厅使用什么样的调控策略。

图中每个模块都通过数据驱动，应用了机器学习算法来实现相应的目标。图中绿色标记是我们已经完成的，标记黄色部分是我们正在或者规划中的工作。在第四章中，我将通过两个例子来介绍我们取得的一些成果。

**2.3 数据监控**

13

人工智能篇



我们还构建了一套策略实时监控系统，方便相关人员查看。例如，图 4 显示的是某个团队的实时压力系数。团队压力系数，我们做到了归一化(所有团队值域一致，含义一致)，并且能根据天气、温度等情况自适应地变化。当前，这个指标我们做到了每隔 5 分钟更新一次。在第四章中，我将会更加详细地介绍团队压力系数的计算方式。

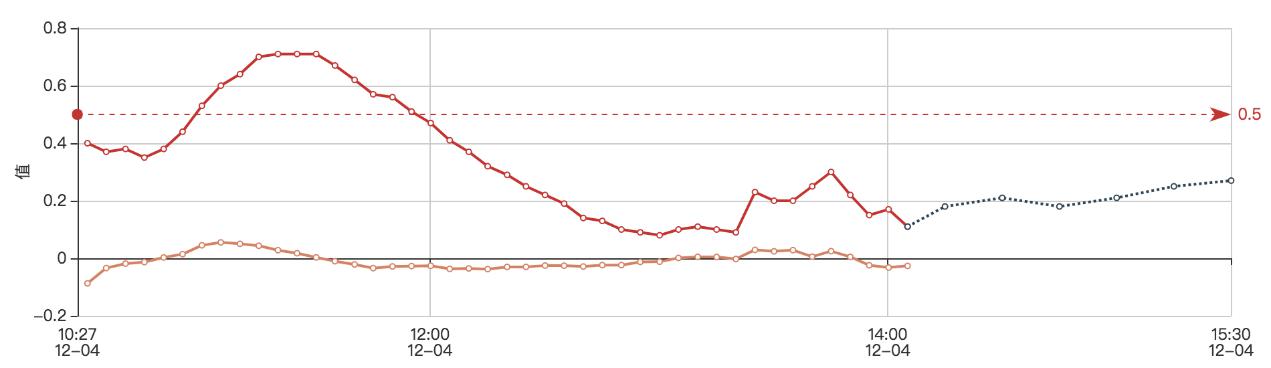


图 4 压力系数监控看板

**三、机器学习算法在压力平衡中的应用**

在这个章节中，我将通过两个案例来介绍机器学习算法在构建饿了么压力平衡系统中起到的关键作用。

**3.1 骑手最大背单能力模型**

骑手的最大背单能力反应了一个骑手的水平，是骑手画像的非常重要的组成部分。第一版，由于我们只是为了得到团队的最大背单能力，所以我们只使用了简单的规则，计算了团队的平均最大背单能力，用这个平均最大背单能力作为这个团队骑手的最大背单能力。显然，这是非常不合理的。

在第二版中，我们通过一些负责的规则，将骑手分成了若干等级，每个等级的骑手具有一个相同的最大背单能力。这个版本的骑手最大背单能力具有了一些个性化，但是任然无法细致、有效地区分不同骑手的水平。

在第三版中，我们采用了机器学习的方法，把这个问题抽象成一个二分类的问题，从而得到了每个骑手在不同背单量情况下超时的概率。图 5 显示了以上介绍的迭代内容，下面介绍 V3.0 版本我们的思路。

14

人工智能篇



图 5 骑手最大背单能力模型迭代

解决这个问题的关键思路在于，我们将骑手最大背单能力的估算转化为一个二分类问题。具体步骤如下：

第一步：我们队每张骑手的运单构建一条训练数据；

第二步：统计这张运单运送过程中骑手的最大背单量作为其中一个特征；第三步：统计其他特征，包括骑手画像相关特征，天气特征等；

第四步：构建二分类的 Label，一张运单超时记为 1，不超时记为 0；第五步：使用大量的训练数据，训练得到一个线性的二分类模型；

第六步：给定一个骑手，固定其他特征，并滑动最大背单量特征，得到该骑手在不同最大背单量下超时的概率。

下面，我详细介绍一下特征工程部分，主要包括以下几块内容：

1）骑手历史最大非超时背单量。包含前 28 天、21 天、14 天、7 天、5 天、3 天的骑手平均最大非超时背单量；

2）天气和温度信息；

1. 骑手的个性化信息：包括骑手的等级，骑手的工作天数，骑手所在团队的信息等；
2. 时间信息：主要包括该日是一周的哪一天，是否是工作日两个特征；

5）运单信息：运单运送过程中骑手的身上最大背单量。

接下来我介绍一下模型的选择和训练集和预测集的构造。根据简单有效原理(即：奥卡姆剃刀定律)，我们采用了常用的线性模型，即 Logistic Regression(LR)模型，我们的考虑如下：

* 线性模型，模型非常简单，有效，计算速度快；
* 模型可解释性很强，便于业务人员理解；
* 问题需要输出[0,1]之间的概率值, LR 使用 sigmoid 函数能够将预测值转化为概率值。

训练集的构造我们采用了最近 14 天的运单数据，即数据日期范围是[T-13,T]， 时间范围限定在了午高峰(10:30~12:30)。由于超时单相比非超时单占比非常小，所以我们对正负样本的比例进行了调整，对非超时样本按照 10%左右进行了降采样。

15

人工智能篇



在构造测试集时，我们对运单运送过程中骑手的最大背单量特征进行滑动，从 0 ~ 30 按照 0.5 的间隔进行构造。然后通过训练的模型进行预测，从而得到骑手在不同背单量下的超时概率。如图 6 所示, 是 taker\_id =342853 的骑手在不同 max\_order\_num\_on\_taker 情况下超时的 probability。

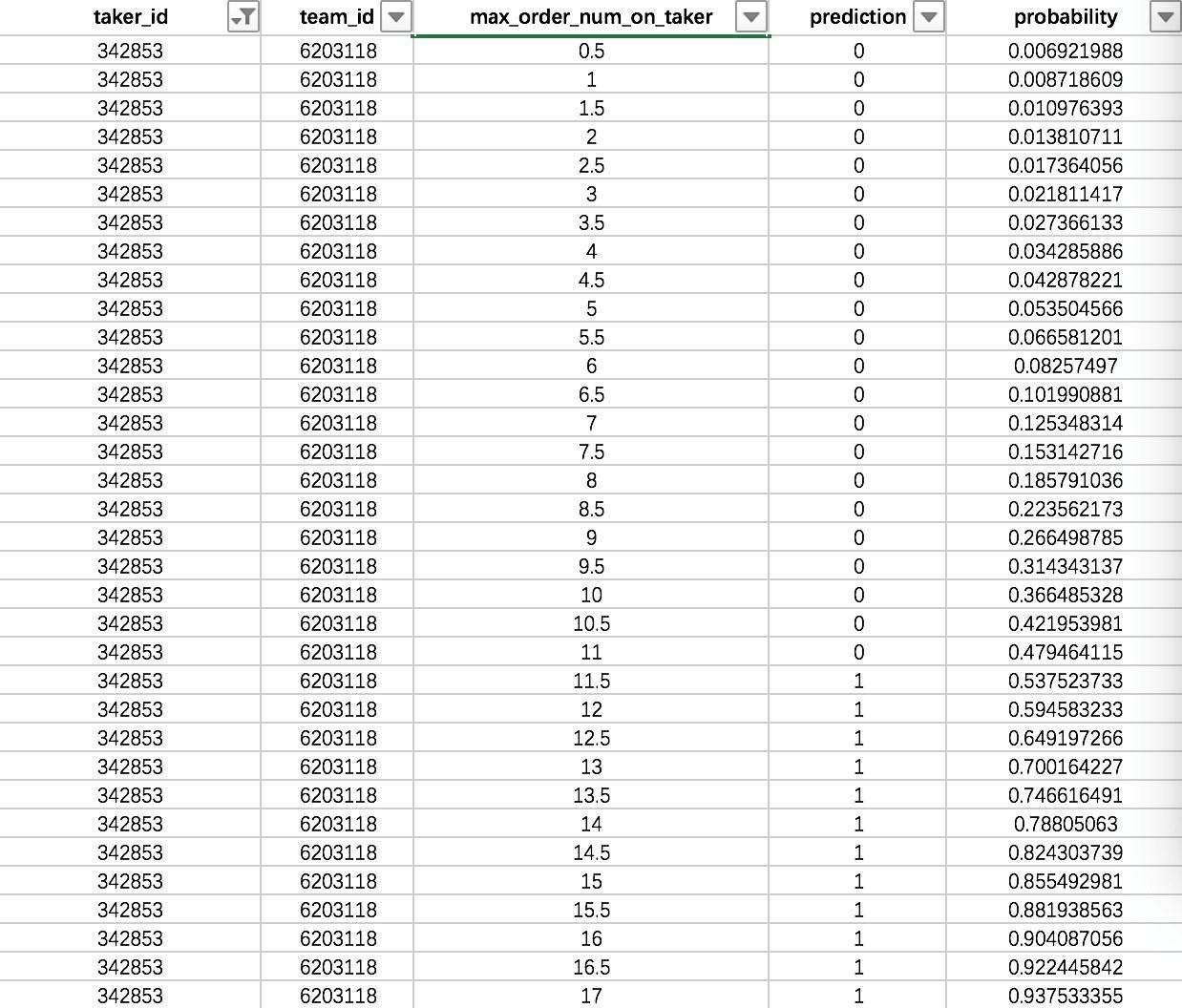


图 6 骑手在不同背单量情况下超时的概率

下面我来介绍一个骑手最大背单模型的重要应用场景，即计算团队的压力系数：

我们定义团队压力系数 = load / (q1 + q2 + .... qn)

1）load=团队负载

2）qn=该团队第 n 位骑手的个性化最大背单量(使用骑手最大背单模型计算的数值)

团队压力系数的定义简洁有效，直接支持了饿了么即时配送相关 10 多个相关业务系统：包括智能派单，压力平衡，运单分流，T 模型，客服系统等。

**3.2 团队压力系数预测模型**

正如章节 3.1 所介绍，团队压力系数是即时配送中一项非常重要的基础指标，因此，如果我们能提前预测这一指标将会对多个业务系统产生巨大的价值。基于此，我们构建了一个实时的团队压力系数预测模型，它能够每隔 5 分钟对团队未来 1 个半小时(每 15 分钟一个时间

16

人工智能篇

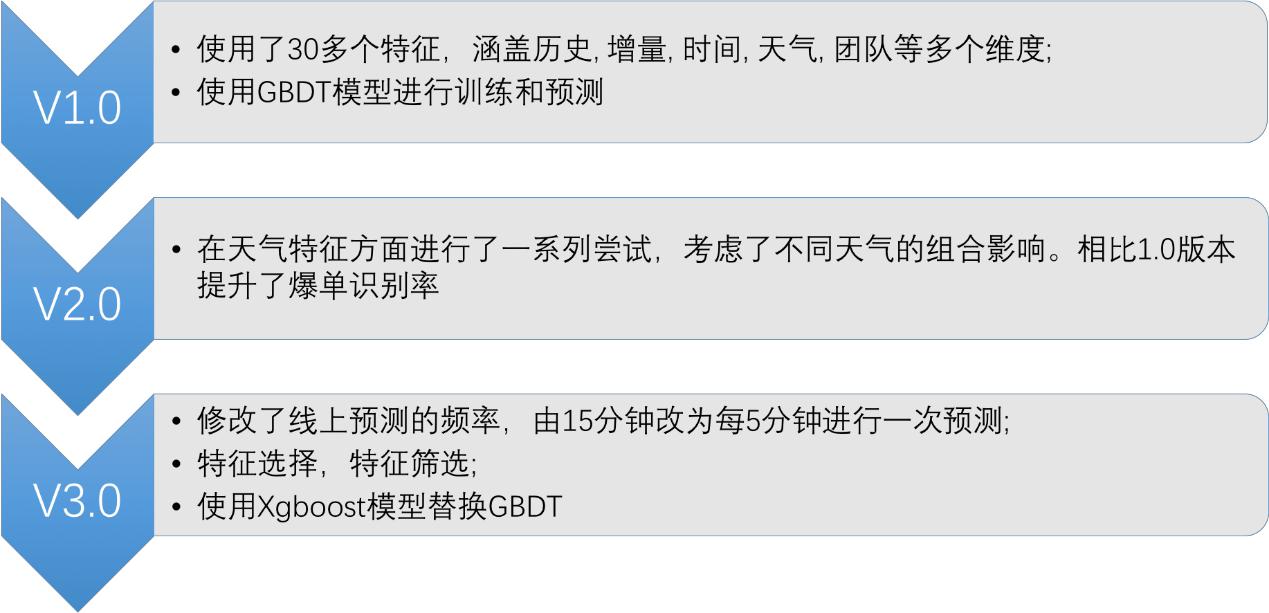


片，共 6 个时间片)的压力系数进行预测。图 4 中的虚线即展示了其中一次预测的值。我们将这个模型应用在以下 2 个场景中：

1）定量描述压力平衡自动调控的目标，辅助确定自动调控的时间点，进行提前调控；

2）应用在智能派单策略上，通过压力系数的预知，规划不同的派单策略。

如下图所示：截止目前，我们完成了团队压力系数预测模型的三轮迭代。通过持续迭代，我们大幅提高了模型预测的准确率和预测的频率。



**四、总结展望**

通过本文，我们希望读者能够对饿了么即时配送体系中的压力平衡系统以及如何利用常见的机器学习算法有效解决 O2O 场景下的实际问题有所了解。展望未来，压力平衡系统中仍然有很多问题可以抽象成机器学习问题（例如：排序学习，时间序列等），希望在不久的未来，有机会和大家分享我们更多的成果。

17

人工智能篇

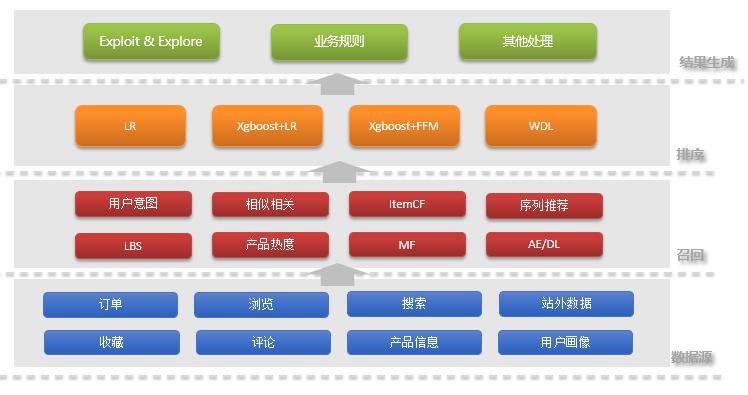


**携程个性化推荐算法实践**

**[作者简介]**携程基础业务研发部-数据产品和服务组，专注于个性化推荐、自然语言处理、图像识别等人工智能领域的先进技术在旅游行业的应用研究并落地产生价值。目前，团队已经为携程提供了通用化的个性化推荐系统、智能客服系统、AI 平台等一系列成熟的产品与服务。

携程作为国内领先的 OTA，每天向上千万用户提供全方位的旅行服务，如何为如此众多的用户发现适合自己的旅游产品与服务，挖掘潜在的兴趣，缓解信息过载，个性化推荐系统与算法在其中发挥着不可或缺的作用。而 OTA 的个性化推荐一直也是个难点，没有太多成功经验可以借鉴，本文分享了携程在个性化推荐实践中的一些尝试与摸索。

推荐流程大体上可以分为 3 个部分，召回、排序、推荐结果生成，整体的架构如下图所示。



召回阶段，主要是利用数据工程和算法的方式，从千万级的产品中锁定特定的候选集合，完成对产品的初步筛选，其在一定程度上决定了排序阶段的效率和推荐结果的优劣。

业内比较传统的算法，主要是 CF[1][2]、基于统计的 Contextual 推荐和 LBS，但近期来深度学习被广泛引入，算法性取得较大的提升，如：2015 年 Netflix 和 Gravity R&D Inc 提出的利用 RNN 的 Session-based 推荐[5]，2016 年 Recsys 上提出的结合 CNN 和 PMF 应用于 Context-aware 推荐[10]，2016 年 Google 提出的将 DNN 作为 MF 的推广，可以很容易地将任意连续和分类特征添加到模型中[9]，2017 年 IJCAI 会议中提出的利用 LSTM 进行序列推荐[6]。2017年携程个性化团队在 AAAI 会议上提出的深度模型 aSDAE，通过将附加的 side information 集成到输入中，可以改善数据稀疏和冷启动问题[4]。

对于召回阶段得到的候选集，会对其进行更加复杂和精确的打分与重排序，进而得到一个更

18

人工智能篇



小的用户可能感兴趣的产品列表。携程的推荐排序并不单纯追求点击率或者转化率，还需要考虑距离控制，产品质量控制等因素。相比适用于搜索排序，文本相关性检索等领域的 pairwise 和 listwise 方法，pointwise 方法可以通过叠加其他控制项进行干预，适用于多目标优化问题。

工业界的推荐方法经历从线性模型＋大量人工特征工程[11] -> 复杂非线性模型-> 深度学习的发展。Microsoft 首先于 2007 年提出采用 Logistic Regression 来预估搜索广告的点击率[12]，并于同年提出 OWLQN 优化算法用于求解带 L1 正则的 LR 问题[13]，之后于 2010 年提出基于 L2 正则的在线学习版本 Ad Predictor[14]。

Google 在 2013 年提出基于 L1 正则化的 LR 优化算法 FTRL-Proximal[15]。2010 年提出的 Factorization Machine 算法 [17] 和进一步 2014 年提出的 Filed-aware Factorization Machine[18]旨在解决稀疏数据下的特征组合问题，从而避免采用 LR 时需要的大量人工特征组合工作。

阿里于 2011 年提出 Mixture of Logistic Regression 直接在原始空间学习特征之间的非线性关系[19]。Facebook 于 2014 年提出采用 GBDT 做自动特征组合，同时融合 Logistic Regression[20]。

近年来，深度学习也被成功应用于推荐排序领域。Google 在 2016 年提出 wide and deep learning 方法[21]，综合模型的记忆和泛化能力。进一步华为提出 DeepFM[15]模型用于替换 wdl 中的人工特征组合部分。阿里在 2017 年将 attention 机制引入，提出 Deep Interest Network[23]。

携程在实践相应的模型中积累了一定的经验，无论是最常用的逻辑回归模型（ Logistic Regression），树模型（GBDT，Random Forest）[16]，因子分解机（FactorizationMachine），以及近期提出的 wdl 模型。同时，我们认为即使在深度学习大行其道的今下，精细化的特征工程仍然是不可或缺的。

基于排序后的列表，在综合考虑多样性、新颖性、Exploit & Explore 等因素后，生成最终的推荐结果。本文之后将着重介绍召回与排序相关的工作与实践。

**一、数据**

机器学习＝数据＋特征＋模型

在介绍召回和排序之前，先简单的了解一下所用到的数据。携程作为大型 OTA 企业，每天都有海量用户来访问，积累了大量的产品数据以及用户行为相关的数据。实际在召回和排序的过程中大致使用到了以下这些数据：

* 产品属性：产品的一些固有属性，如酒店的位置，星级，房型等。
* 产品统计：比如产品一段时间内的订单量，浏览量，搜索量，点击率等。
* 用户画像：用户基础属性，比如年纪，性别，偏好等等。
* 用户行为：用户的评论，评分，浏览，搜索，下单等行为。

19

人工智能篇



值得注意的是，针对统计类信息，可能需要进行一些平滑。例如针对历史 CTR 反馈，利用贝叶斯平滑来预处理。

**二、召回**

召回阶段是推荐流程基础的一步，从成千上万的 Item 中生成数量有限的候选集，在一定程度上决定了排序阶段的效率和推荐结果的优劣。而由 OTA 的属性决定，用户的访问行为大多是低频的。这就使得 user-item 的交互数据是极其稀疏的，这对召回提出了很大的挑战。在业务实践中，我们结合现有的通用推荐方法和业务场景，筛选和摸索出了几种行之有效的方法：

**Real-timeIntention**

我们的实时意图系统可以根据用户最近浏览下单等行为，基于马尔科夫预测模型推荐或者交叉推荐出的产品。这些候选产品可以比较精准的反应出用户最近最新的意愿。

**BusinessRules**

业务规则是认为设定的规则，用来限定推荐的内容范围等。例如机票推酒店的场景，需要通过业务规则来限定推荐的产品只能是酒店，而不会推荐其他旅游产品。

**Context-Based**

基于 Context 的推荐场景和 Context 本身密切相关，例如与季候相关的旅游产品（冬季滑雪、元旦跨年等）。

20

人工智能篇



**LBS**

基于用户的当前位置信息，筛选出的周边酒店，景点，美食等等，比较适用于行中场景的推荐。地理位置距离通过 GeoHash 算法计算，将区域递归划分为规则矩形，并对每个矩形进行编码，筛选 GeoHash 编码相似的 POI，然后进行实际距离计算。

**CollaborativeFiltering**

协同过滤算法是推荐系统广泛使用的一种解决实际问题的方法。携程个性化团队在深度学习与推荐系统结合的领域进行了相关的研究与应用，通过改进现有的深度模型，提出了一种深度模型 aSDAE。该混合协同过滤模型是 SDAE 的一种变体，通过将附加的 side information 集成到输入中，可以改善数据稀疏和冷启动问题，详情可以参见文献[4]。

**SequentialModel**

现有的矩阵分解(Matrix Factorization)方法基于历史的 user-item 交互学习用户的长期兴趣偏好，Markov chain 通过学习 item 间的 transition graph 对用户的序列行为建模[3]。事实上，在旅游场景下，加入用户行为的先后顺序，从而能更好的反映用户的决策过程。我们结合 Matrix Factorization 和 Markov chain 为每个用户构建个性化转移矩阵，从而基于用户的历史行为来预测用户的下一行为。在旅游场景中，可以用来预测用户下一个目的地或者 POI。

除此之外，也可以使用 RNN 来进行序列推荐，比如基于 Session 的推荐[5]，使用考虑时间间隔信息的 LSTM 来做下一个 item 的推荐等[6]。

21

人工智能篇



此外，一些常见的深度模型(DNN, AE,CNN 等)[7][8][9][10]都可以应用于推荐系统中，但是针对不同领域的推荐，需要更多的高效的模型。随着深度学习技术的发展，相信深度学习将会成为推荐系统领域中一项非常重要的技术手段。以上几种类型的召回方法各有优势，在实践中，针对不同场景，结合使用多种方法，提供给用户最佳的推荐，以此提升用户体验，增加用户粘性。

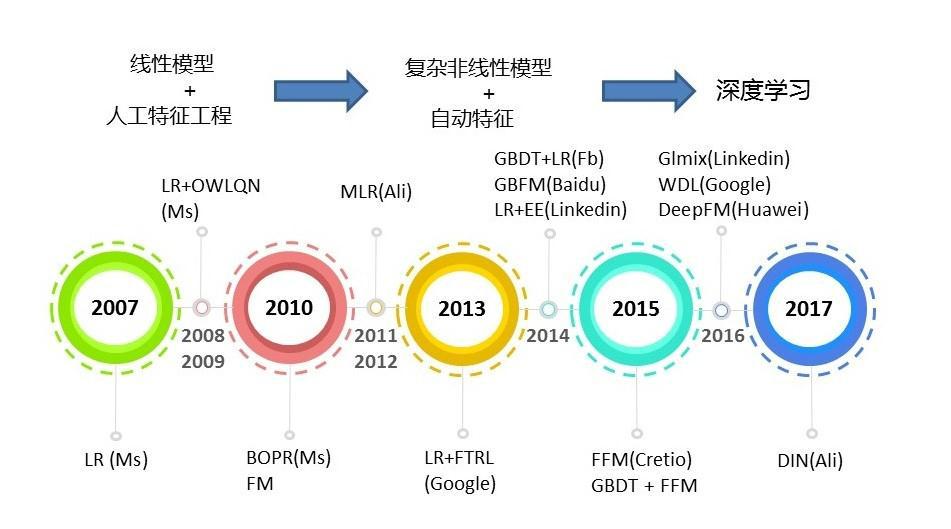
**三、排序**

以工业界在广告、搜索、推荐等领域的实践经验，在数据给定的条件下，经历了从简单线性模型＋大量人工特征工程到复杂非线性模型＋自动特征学习的演变。在构建携程个性化推荐系统的实践过程中，对于推荐排序这个特定问题有一些自己的思考和总结，并将从特征和模型这两方面展开。

**Model**

个性化排序模型旨在利用每个用户的历史行为数据集建立其各自的排序模型，本质上可以看作多任务学习(multi-task learning)。事实上，通过加入 conjunctionfeatures，也就是加入 user 和 product 的交叉特征，可以将特定的 multi-task 任务简化为单任务模型。

梳理工业界应用的排序模型，大致经历三个阶段，如下图所示：



本文并不准备详细介绍上图中的算法细节，感兴趣的读者可以查看相关论文，以下几点是我们的一些实践经验和体会。

1）在实践中选用以 LR 为主的模型，通过对数据离散化、分布转换等非线性处理后使用 LR。一般的，采用 L1 正则保证模型权重的稀疏性。在优化算法的选择上，使用 OWL-QN 做 batch learning，FTRL 做 online learning。

22

人工智能篇



2）实践中利用因子分解机（FactorizationMachine）得到的特征交叉系数来选择喂入 LR 模型的交叉特征组合，从而避免了繁杂的特征选择工作。一般的受限于模型复杂度只进行二阶展开。对于三阶以上的特征组合可以利用基于 mutual information 等方法处理。已有针对高阶因子分解机（HighOrder FM）的研究，参见文献[24]。

3）对于 Wide and Deep Learning，将 wide 部分替换 gbdt 组合特征，在实验中取得了较好的效果，并将在近期上线。后续的工作将针对如何进行 wide 部分和 deep 部分的 alternatingtraining 展开。

**Feature Engineering**

事实上，虽然深度学习等方法一定程度上减少了繁杂的特征工程工作，但我们认为精心设计的特征工程仍旧是不可或缺的, 其中如何进行特征组合是我们在实践中着重考虑的问题。一般的，可以分为显式特征组合和半显式特征组合。

**显式特征组合**

对特征进行离散化后然后进行叉乘，采用笛卡尔积(cartesian product)、内积(inner product) 等方式。

在构造交叉特征的过程中，需要进行特征离散化；针对不同的特征类型，有不同的处理方式。

1）numerical feature

无监督离散化：根据简单统计量进行等频、等宽、分位点等划分区间有监督离散化：1R 方法，Entropy-BasedDiscretization (e.g. D2，MDLP)

2）ordinal feature（有序特征）

编码表示值之间的顺序关系。比如对于卫生条件这一特征，分别有差，中，好三档，那么可以分别编码为(1,0,0),(1,1,0),(1,1,1)。

3）categorical feature (无序特征)



* 离散化为哑变量，将一维信息嵌入模型的 bias 中，起到简化逻辑回归模型的作用，降低了模型过拟合的风险。
* 离散特征经过 OHE 后，每个分类型变量的各个值在模型中都可以看作独立变量，增强

23

人工智能篇



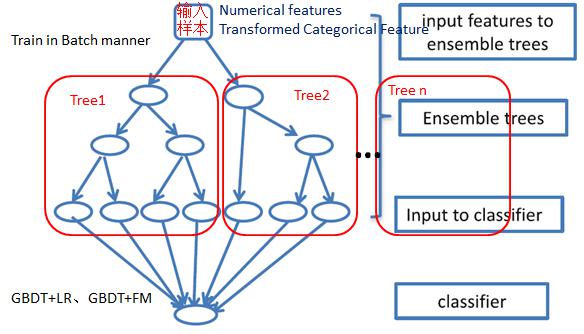
拟合能力。一般的，当模型加正则化的情况下约束模型自由度，我们认为 OHE 更好。

* 利用 feature hash 技术将高维稀疏特征映射到固定维度空间

**半显式特征组合**

区别于显式特征组合具有明确的组合解释信息，半显式特征组合通常的做法是基于树方法形成特征划分并给出相应组合路径。

一般做法是将样本的连续值特征输入 ensemble tree，分别在每颗决策树沿着特定分支路径最终落入某个叶子结点得到其编号，本质上是这些特征在特定取值区间内的组合。ensemble tree 可以采用 Gbdt 或者 random forest 实现。每一轮迭代，产生一棵新树，最终通过 one-hotencoding 转化为 binary vector，如下图所示。



以下几点是我们在实践中的一些总结和思考。

1）在实验中发现如果将连续值特征进行离散化后喂入 gbdt，gbdt 的效果不佳，AUC 比较低。这是因为 gbdt 本身能很好的处理非线性特征，使用离散化后的特征反而没什么效果。 xgboost 等树模型无法有效处理高维稀疏特征比如 user id 类特征，可以采用的替代方式是:将这类 id 利用一种方式转换为一个或多个新的连续型特征，然后用于模型训练。

2）需要注意的是当采用叶子结点的 index 作为特征输出需要考虑每棵树的叶子结点并不完全同处于相同深度。

3）实践中采用了 Monte Carlo Search 对 xgboost 的众多参数进行超参数选择。

4）在离线训练阶段采用基于 Spark 集群的 xgboost 分布式训练，而在线预测时则对模型文件直接进行解析，能够满足线上实时响应的需求。

24

人工智能篇



此外，在实践发现单纯采用 Xgboost 自动学到的高阶组合特征后续输入 LR 模型并不能完全替代人工特征工程的作用；可以将原始特征以及一些人工组合的高阶交叉特征同 xgboost 学习到的特征组合一起放入后续的模型，获得更好的效果。

**四、总结**

完整的推荐系统是一个庞大的系统，涉及多个方面，除了召回、排序、列表生产等步骤外，还有数据准备与处理，工程架构与实现，前端展现等等。在实际中，通过把这些模块集成在一起，构成了一个集团通用推荐系统，对外提供推服务，应用在 10 多个栏位，60 多个场景，取得了很好的效果。本文侧重介绍了召回与排序算法相关的目前已有的一些工作与实践，下一步，计划引入更多地深度模型来处理召回与排序问题，并结合在线学习、强化学习、迁移学习等方面的进展，优化推荐的整体质量。

**References**

1. Koren, Yehuda,Robert Bell, and Chris Volinsky. "Matrix factorization techniques forrecommender systems." Computer 42.8 (2009).
2. Sedhain, Suvash,et al. "Autorec: Autoencoders meet collaborative filtering." Proceedingsof the 24th International Conference on World Wide Web. ACM, 2015.
3. Rendle, Steffen,Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." Proceedings ofthe 19th international conference on World wide web. ACM, 2010.
4. Dong, Xin, etal. "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems." AAAI. 2017.
5. Hidasi, Balázs,et al. "Session-based recommendations with recurrent neuralnetworks." arXiv preprint arXiv:1511.06939 (2015).
6. Zhu, Yu, et al."What to Do Next: Modeling User Behaviors by Time-LSTM." Proceedingsof the Twenty-Sixth International Joint Conference on Artificial Intelligence,IJCAI-17. 2017.
7. Barkan, Oren,and Noam Koenigstein. "Item2vec: neural item embedding for collaborative filtering." Machine Learning for Signal Processing (MLSP), 2016 IEEE 26thInternational Workshop on. IEEE, 2016.
8. Wang, Hao,Naiyan Wang, and Dit-Yan Yeung. "Collaborative deep learning forrecommender systems." Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015.
9. Covington, Paul,Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM Conference on RecommenderSystems. ACM, 2016.
10. Kim, Donghyun, et al. "Convolutional matrix factorization fordocument context-aware recommendation." Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016.[11] https://mli.github.io/2013/03/24/the-end-of-feature-engineering-and-linear-model/[12] Richardson, Matthew, Ewa Dominowska, and Robert Ragno."Predicting

clicks: estimating the click-through rate for new ads."Proceedings of the 16th international conference on World Wide Web. ACM, 2007[13] Andrew, Galen, and Jianfeng Gao. "Scalable

25

人工智能篇



training of L1-regularized log-linear models." Proceedings of the 24th international conference on Machine learning. ACM, 2007.[14] Graepel, Thore, et al. "Web-scale bayesian click-through rateprediction for sponsored search advertising in microsoft's bing searchengine." Omnipress, 2010.

1. McMahan, H. Brendan, et al. "Ad click prediction: a view fromthe trenches." Proceedings of the 19th ACM SIGKDD international conferenceon Knowledge discovery and data mining. ACM, 2013.
2. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable treeboosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
3. Rendle, Steffen. "Factorization machines." Data Mining(ICDM), 2010 IEEE 10th International Conference on. IEEE, 2010.
4. Juan, Yuchin, et al. "Field-aware factorization machines for CTR prediction." Proceedings of the 10th ACM Conference on RecommenderSystems. ACM, 2016.
5. Gai, Kun, et al. "Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction." arXiv preprint arXiv:1704.05194(2017).
6. He, Xinran, et al. "Practical lessons from predicting clickson ads at facebook." Proceedings of the Eighth International Workshop onData Mining for Online Advertising. ACM, 2014.
7. Cheng, Heng-Tze, et al. "Wide & deep learning forrecommender systems." Proceedings of the 1st Workshop on Deep Learning forRecommender Systems. ACM, 2016.
8. Guo, Huifeng, et al. "DeepFM: A Factorization-Machine basedNeural Network for CTR Prediction." arXiv preprint arXiv:1703.04247(2017).
9. Zhou, Guorui, et al. "Deep Interest Network for Click-Through Rate Prediction." arXiv preprint arXiv:1706.06978 (2017).
10. Blondel, Mathieu, et al. "Higher-order factorization machines." Advances in Neural Information Processing Systems. 2016.
11. http://breezedeus.github.io/2014/11/20/breezedeus-feature-hashing.html
12. https://en.wikipedia.org/wiki/Categorical\_variable
13. https://www.zhihu.com/question/48674426
14. 多高的 AUC 才算高？https://zhuanlan.zhihu.com/p/24217322

26

人工智能篇



**平安银行算法实践**

**[作者简介]**潘鹏举,平安银行大数据平台AI算法和分析团队负责人。2017年加入平安，组建 AI 和算法团队，推动 AI 在银行业务的应用。此前曾在携程任职，撸代码、写文档、出规范、带团队，参与设计算法工程化架构，带领算法团队助力酒店服务提升。本文来自潘鹏举在第二届携程云海机器学习沙龙上的分享。

**背景**

银行是偏传统的行业，目前正在遭受互联网和 P2P 等公司的竞争压力，所以我们正在进行零售转型，拥抱互联网和金融科技。在最近不到一年的时间里，我们在算法方面做了一些尝试和探索，并对未来的一些算法应用有一些思考。整体来说，今天我想分享的内容主要分三大块：

1、业务背景介绍

2、算法实践分享

3、一些思考

**一、业务背景**

首先来看银行的核心 KPI。从下图中可以知道，银行核心的 KPI 有两个：AUM 和 LUM。



AUM 表示的是在管资产，包括了一些存款、大额存单、同业拆借、央妈给银行代管的钱，表示的是银行从其他渠道拿到的钱；LUM 表示的是借出资金，信用卡也是其中一种类别，表示的是银行借给其他人、政府或者机构的金额。只要借出资产的利率超过借入资产的利率，那么这笔资金的转换就是有收益的。

27

人工智能篇



在这两个资产汇总，在管资产是重中之重，因为国家有规定，LUM/AUM 应该小于某个比例，所以银行资产规模的大小其实是由 AUM 决定的。LUM 相对来说是比较容易达成，需要钱的个人、机构是很多的，但因为风控方面的考虑同时很多人没有人行征信报告，所以正常情况下从银行申请贷款是件非常痛苦的事情，也正因为如此，现如今 P2P 行业、消费金融行业层出不穷，因为供给和需求的不对等催生了如今的现象。

从上面的业务背景描述，我们可以看到 AUM 的核心是客群管理，因为我们在 LUM 方面比较谨慎，换位一下，银行要从其他渠道获取资金也是非常艰难的。LUM 的核心是风控，坏账率是需要关注的一个指标，当然坏账率高低和利差高低有直接关系，这也就是为什么现在的消金公司的逾期率和坏账率比银行高很多的原因。以上两点之外，我们也可以了解到银行受到的一些挑战：1. 受监管 2. 合规性 3.获客难 4. 数据稀疏。

受监管是因为平安银行属于全国性商业银行，受到国家政策的限制和监管，所做的一切都是合规性先行，我们不能做非政府允许或者非个人授权的事情。数据稀疏是针对老客户和新客户来说，老客户因为我们的产品的频次较低，所以能够获取的用户数据较少。而新客户因为我们可以获取的数据量较少，所以我们经常会跟外部的供应商进行合作。在这点上，其实是个机会，所以针对银行、保险和证券这部分的金融领域公司来说，外部的数据供应商是有一定的机会的。

面对这些挑战，我们也尝试在用一些算法实践来帮助提升银行的 AUM 或者降低风控风险。

**二、算法实践**

在说算法实践之前，我们整体的数据应用的架构如下。分别是基础设施、数据层、应用层三部分。



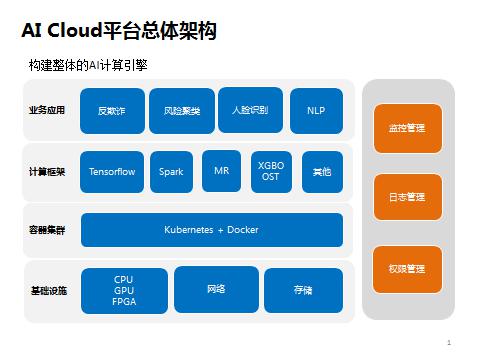
数据服务层主要提供的一些通用的产品和服务，而最底层的基础设施层，我们使用的现在常用的大数据技术组件，比如 hadoop，hive，spark，ES，Redis 等组件。

28

人工智能篇



这里说一下 AI 基础能力这块，说的是专门针对的算法的一个平台级别的设计，其中核心思想是基于 Docker 来进行算法方面的封装和部署应用，达到一键部署和训练的目的。目前我们正在自研这个平台，如果未来设计完成了，可以针对这部分内容进行专题分享。基于新型的架构设计，我们大大的加快算法的迭代速度，加快产出。



之前提到了一些架构面的事情，现在重点说一下我们在算法应用的探索，我们这边整体的应用概览如下：



从此图中，我们的 AI 能力建设大致区分成了深度学习和算法应用这两块。应用层面中机器学习主要说的算法应用，结合业务的场景进行的算法的应用，帮助提升智能化和自助化。知识图谱是针对现有知识的梳理和串联，这部分数据在风控方面用处很多，我们现在在尝试串联一切，通过图谱的方式来去发现数据不一致的地方。

29

人工智能篇



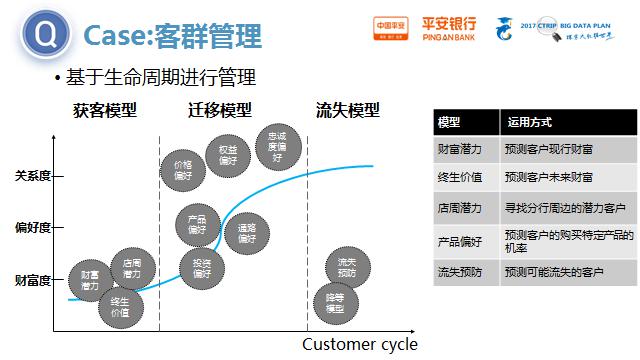
图像、语音和文本是根据现在业务存在的大量人工的事情，可以借助现在新型的技术和方法来提高自动化。文本这块，我们服务于 app 搜索和智能客服这两块。这里提一下文本这块，因为是场景化的需求居多，所以我们会重点发展自有的技术。

关于算法应用这块，我们的原则是：如果利用一些开源的技术短时间内也和市场主流的产品有一定的差距，那么这部分技术可以采用采购的方法来短时间内达到市场主流竞争中去。举个例子，语音的 ASR 方面，科大讯飞的效果最高那么我们就会直接采用讯飞的技术进行 ASR 转换，但是在语音前处理和后处理方面，我们就会自研，因为在语音方面的去噪和增强方面，很多供应商是要做定制化开发的，对于定制化开发这部分，我们会自研，否则我们就只能一直依赖外部供应商，这个从长远来说，对我们是不利的。

说完了面上的应用。我们来看一些具体的例子，有助于理解我们在某些应用是怎么思考的。

先看第一个例子，关于客群管理的。

**案例 1：客群管理**

****

从上图可以看到，我们的客群是根据客户的生命周期来进行分析和管理的。横轴是客户的生命周期，从获客、迁移到流失，纵轴是业务的细分维度，这样交叉可以区分出不同的客群出来。根据不同的客群我们会给出不同的 offer 和权益，不同的权益享受不同的待遇。

在客群管理中，我们会建不同的子模型用来对客户进行精准化运营。这些不同的子模型，我们的落地方案基本是走客户画像这套体系，落地在标签系统里面，这样其他人想调用的时候都可以直接拉出相应的结果出来。关于客群经营这块，其实很多公司都有涉及，但是都没有深入挖掘并细化，没有形成一整套的解决方案，都是各个部门各自为战。

根据我们的测试上来说，通过精细化运营的方式比传统的粗放式运营方式，成本和效益最少会提高 5 倍。我们银行也在积极的进行探索，希望能够用更加智能化的方式来提高运营，提升业务产出。

30

人工智能篇



**案例 2：客户画像**

接下来我们看一下客户画像如何赋能到一线人员。下图是我们在给一线人员开发的客群分析的界面。



可以从上图看到，我们提供了不同的数据职能模块，前面是一些事实性的标签数据，用来表示当前客户的一些基本信息，后面是一些算法给出的推荐和预测结果。

其中需要留意的是，在预测类标签中，我们额外提供了很多的辅助决策类的数据，比如流失预测概率中，我们会提供计算流失概率的重要因素，把这些重要因素展示给一线人员，告诉他们什么因素导致此客户的流失概率较高。同时，我们也会提供模型预估的流失准确率和实际准确率的比较结果，用来发现当前模型是否有比较显著的下降，以方便我们及时的进行模型的更新。

从这个案例中，告诉我们赋能一线的时候，不仅仅需要提供精准预测的结果，还需要提供更多的决策依据，否则无法指导执行的人进行有针对性的改善。当然，提供辅助决策的前提是预测结果是不是直接可以用来决策。接下来我们来看一下直接利用模型结果决策的例子。

**案例 3：业务预测**

我们接着看比较常规的业务预测

31

人工智能篇



这个是某业务量预测需求，项目背景是去预测未来 3 天的业务量，根据这个业务量我们会进行排班的设计，所以这个需求是直接用预测结果进行决策的例子。

针对这个项目背景和需求，我们先拉取了数据来进行分析，发现历史数据不全，能拉取到的数据就不到半年，所以周期性的规律我们都没有办法捕捉。所以我们尝试了右边的模型融合的方法，尝试了不同的预测方法进行预测，然后再结合规则进行了最终模型结果的输出。

其中历史同天平均值表示的是最近三个月的同一天的业务量预测平均值。

其中规则的方法针对的是月初的情况，我们发现月初的结果和其他的走势很不一样，所以我们针对月初使用了一些固定的规则来进行预测。

通过以上方法，我们可以把预测的绝对偏差控制在 9%以下，在数据量如此小的情况下能达到如此精度，我们觉得还是做的不错的。也在此把方法分享给大家，看一下对大家有没有一些启发。

说完了一些预测类的事情，我们接下来说一下我们在图谱方面的尝试，这是个体力活。

**案例 4：图谱**

32

人工智能篇

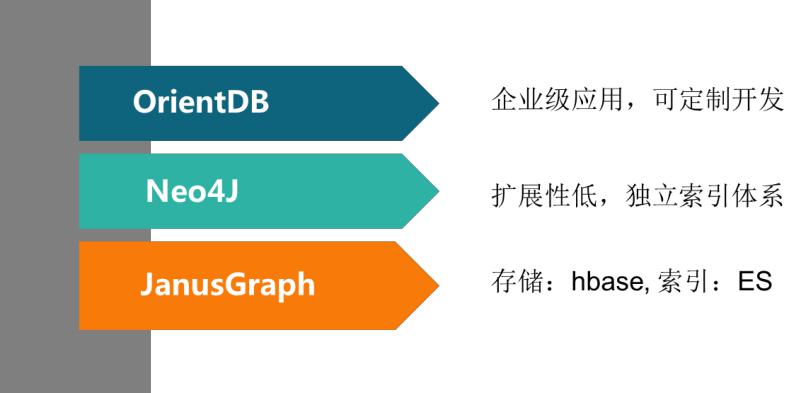


上图是我们在图谱方面做的两个尝试，左边的这张图是卡号、进件号码和激活号码之间的聚类结果，我们根据这个聚类结果发现了一些疑似薅羊毛的团伙，并针对这些团伙进行定点的分析，比如地域，发现的结果还是蛮有意思的。

右边这个图是我们根据左图的测试案例进行外衍生出来的知识图谱结果，这个是我们的一个数据产品。我们可以利用这个产品查询到不同电话之间有关系的人、不同卡号之间有关系的人出来，这个产品在对某些风险案件的反查和一些新的规则的发现是有帮助的。

在对图谱的存储上，我们尝试部署了一些图数据库，比如 neo4j、OrientDB 和 JanusGraph。

对比结果如下：



OrientDB 和 JanusGraph，我们都在一定的尝试，目前在使用 OrientDB 做了一些地址方面的 POI 存储，用来进行多个三元组的存储。

关于图数据库上，它是一个存储介质。在图挖掘上，我们认为重要的是根据场景来进行分析和挖掘，所以多做一些数据分析和探索是重中之重，存储只是解决了快速部署应用的问题。

我们希望在未来有更多的图谱方面的挖掘和应用，万物串联是我们在做各种应用的基础能力。

33

人工智能篇



**三、一些思考**

在我们做一些应用实践的时候，我们也有一些感悟，这些感悟对我们加快数据赋能公司也有一些帮助。

第一个感悟是：

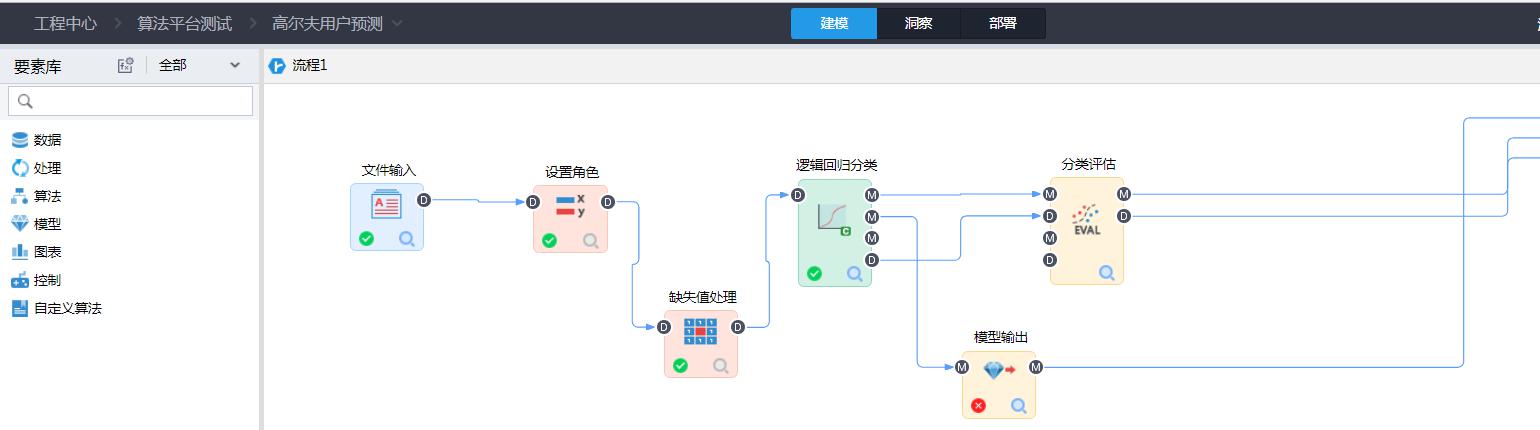
在很多算法落地的场景中，我们都比较需要数据产品经理的角色，比如客户画像、客群管理、业务预测等，数据产品经理会梳理好业务和数据对接的场景，让算法工程师的工作职能变得相对专一一点，他只要了解业务，数据梳理并把模型开发上线就可以，而在最终的页面展示和业务系统对接和沟通、协调上面会由数据产品经理去完成，他们同时会兼任一些项目管理的工作。

通过这个分工，可以让整个算法项目进度完成的更加顺畅。我以前带团队的时候，这个角色一般是由我来承担的，即分组经理承担，但是分组经理因为顾及的面较多，涉及到多个项目的跟进和资源协调，所以在单个项目完成度上打了一定的折扣。

现在，通过新增的数据产品经理角色，可以加快数据落地和闭环过程。算法应用人员、数据产品经理、分组经理。外部系统开发各司其职，很好的完成了数据赋能的任务。

第二个感悟是：

我们需要一个统一的 AI 算法平台，集模型训练、部署、可视化于一体。通过这个平台可以减少很多不同算法工程师的重复工作。

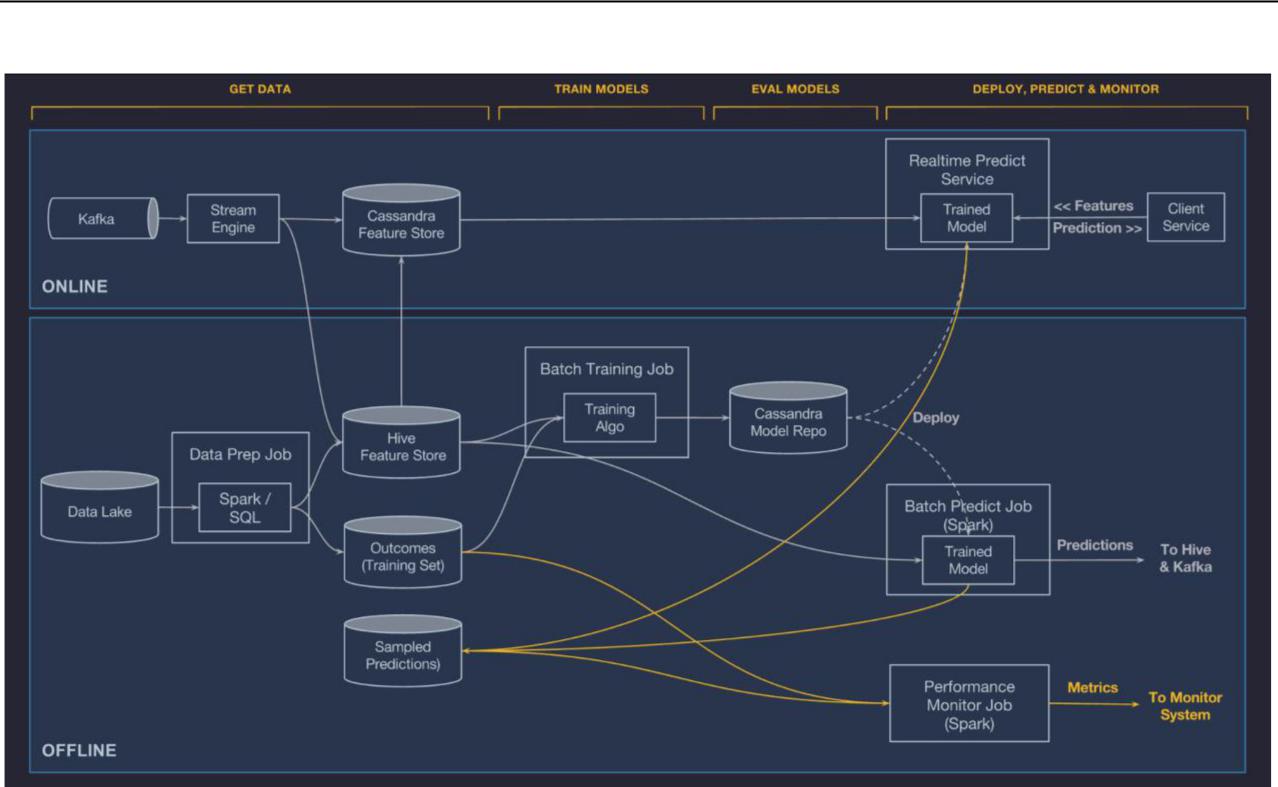


上图是我们目前使用的一个算法平台，挺好看的，但是功能上需要完善很多，所以我们也在设计一个新的算法平台，用来减轻一些重复工作量。现在外界也有很多公司也在开发相应的产品，其实目的就是想通过减少使用算法的门槛来提高算法在公司的使用情况。从目前来看，很多算法的壁垒和门槛已经慢慢的变低了，所以未来对算法工程师的要求就更加严格了。

最后，说一下理想中的算法平台可以参考一下 UBER 设计的平台模式，从数据源头到部署运维一体化，覆盖了 online 和 offline 渠道。

34

人工智能篇



35

人工智能篇



**携程“小诗机”背后的机器学习和自然语言处理技术**

**[作者简介]**孙玉霞，携程技术中心基础业务部算法工程师，南京师范大学硕士，主要研究方向为自然语言处理，参与小诗机，智能客服，产品强化等相关项目。

2017 年年初，携程推出了“诗情画意”小诗机，让机器能够“理解”，“欣赏”用户拍摄的照片，并基于小诗机自有的庞大知识库体系，写出符合图片的意境和内容的古诗。

目前，小诗机和上海地区诗人盲测实验结果显示，小诗机已经达到人类诗人的水平，专业评委和大众评委无法区分出哪些是小诗机的“大作”，整体评分排名靠前。

同时，小诗机还具有基于图片检索古诗，作藏头诗，宝塔诗等功能。小诗机是人工智能在人类创造力和理解力上的挑战，让人们在旅游的同时拥有诗和远方。

运用了知识图谱和图片识别、自然语言深度学习方面的前沿技术，是大数据和人工智能的结合产物。我们使用机器学习进行景点实体关系和特征的提取，构建大规模的旅游景点知识图谱，包括全球数万景点、地区、美食、天气等数据；使用自然语言技术基于 30 多万首古今诗篇进行语义层级的主题诗歌自动生成。强大的知识支撑和先进技术的融合造就了“诗情画意”的小诗机。部分样例如下：

基于景物识别结果成诗。如图 1，图片包括楼，林木，梅花等元素，成诗时结合了当时的季节天气等信息，如日色，冬晴光。



36

人工智能篇



图 1 景物识别信息成诗

基于人物性别，年龄，表情等成诗。如图 2，图片中是一个表情严肃的年轻男人。



图 2 人物识别信息成诗

基于景点信息成诗。如图 3，成诗结果融合了西湖的特色和周边景点，比如曲院，孤山，寒湖，莲叶，茶园，寺庙等。



图 3 景点信息成诗

为了评估小诗机的效果，我们邀请 5 位文学爱好者（高校文学院学生或者老师，具有较好的文学功底）跟小诗机进行 PK：基于 11 组主题赋诗，文学爱好者有一天的时间进行诗歌创

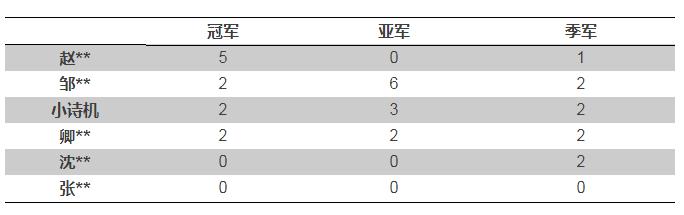
37

人工智能篇



作，而小诗机即兴创作，同时邀请两位专业诗人和四位大众诗人作为评委。11 个主题的评比结果如表 1，小诗机在 2 个主题下获得第一名，3 个主题下获得第二名，2 个主题下获得第三名，综合排名第三，通过 了诗歌创作的“图灵测试”。

表 1 基于 11 场次的评比结果



下面我们通过整体流程介绍、知识图谱构建、图片识别、成诗引擎这四个方面来揭开“小诗机”的面纱。

**一、整体流程**

小诗机的基本流程如图 4，主要包括知识图谱、图片识别和写诗引擎。

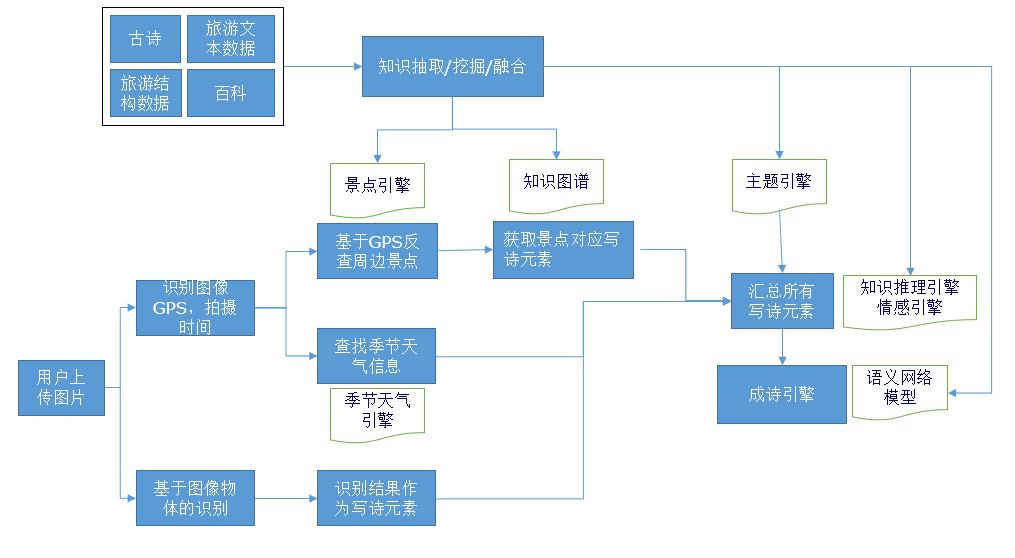


图 4 小诗机的整体流程

**二、知识图谱构建**

**2.1 知识来源**

为了构建全面的旅游行业知识图谱，我们融合了多个维度的知识源，一方面是携程特有的大

38

人工智能篇



量的旅游行业的数据和景点数据，另一方面是互联网上关于城市、历史、传说、民俗方面的数据，具体分为如下三个部分：

非结构化数据： 携程的景点简介，用户游记，用户评论等文本信息。

半结构化数据：以维基百科，百度百科为代表的大规模知识库，包含了大量结构化和半结构化数据，可以高效地应用到知识图谱中。

结构化数据：包括自由行，团队游，酒店，景点，用户意图等较为全面的结构化旅游数据。

针对结构化数据，除了使用一些相关的维度信息，如景点所属的城市，同时也会进行一些相关的统计分析，实现深入的信息提取和筛选，如对每个城市下重要的景点进行排名。

**2.2 知识抽取**

文本数据的知识抽取会涉及到许多自然语言处理的基本技术，包括分词，词性标注，依存句法分析，语义角色标注，命名实体识别等，实现对文本数据的清洗和结构化。主要的提取目标包括如下几个部分：

实体抽取：利用 CRF++和字典结合的方式，从文本中自动识别人名，地名（国家，省市州，城市，景点），机构，节气，时间等实体[16]。

关系抽取：主要利用一些启发式的规则和半监督的方法来从新学到的实例中学习新的 pattern 并扩充 pattern 集合挖掘各个实体之间的关系。

主题抽取：为了构建更全面的旅游知识体系，同时使用改进的 tfidf，卡方 + TextRank，LDA等方法[17]综合进行文章主题，关键词，摘要的抽取，从而实现景点实体和特征实体的关系挖掘。

**2.3 知识融合**

我们的数据来自于不同的数据源，需要对各个数据源提取出来的知识进行融合。实体的融合和匹配是多数据源融合的核心，一方面通过语义向量维度/字面维度的相似度来进行实例、属性、概念等的匹配[19][20]，另一方面结合实体对应的相关属性以及结构信息进行多维度多层次的匹配[18]，最终利用自定义权重进行融合。

**2.4 知识推理**

在构建的实体关系库上，使用基于符号逻辑推理方法，推理出新的实体关系以及约束关系。

**三、图片识别**

对于计算机视觉任务而言，大多模型都是基于卷积神经网络（Convolutional Neural Network, CNN）。LeNet5[12]是最早的卷积神经网络，但是由于硬件的限制并没有被广泛的采用。在视

39

人工智能篇



觉领域竞赛 ILSVRC 2012 上，AlexNet[14]的出现使得图片分类的正确率得到了大幅度的提升，从传统的 74.2%提升到 83.6%，它是 LeNet5 的一种更深更宽的版本，首次利用 GPU 进行加速运算，并成功应用了 ReLU、Dropout 和 LRN 等技巧。在此基础上，后续出现了 VGGNet[13]、GoogleNet[11]、ResNet[15]等，网络的深度和宽度不断提升，准确率也达到并超过了人类肉眼辨识的水平。

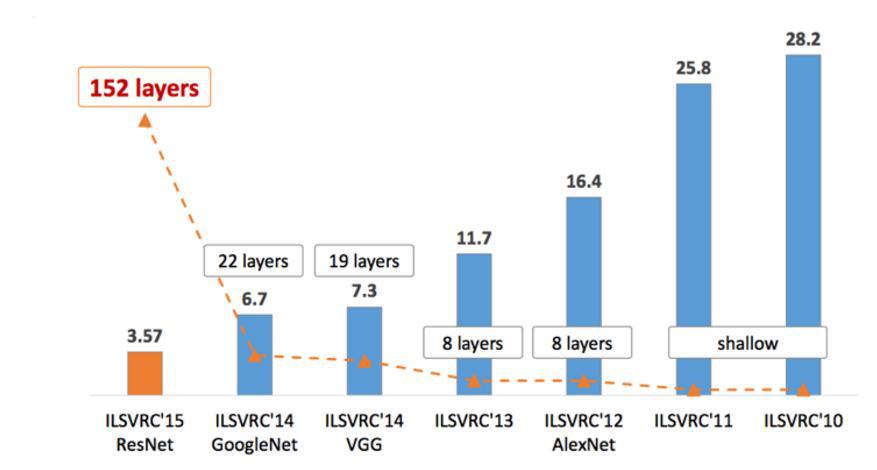


图 5 ILSVRC 历年的 Top-5 错误率

计算机视觉识别领域具有大量的公开数据集，这些数据集大部分具有高质量的标注。迁移学习能很好地运用这些数据，让我们在拥有少量标注数据的情况下取得较好的图片识别结果[1]。

使用 inception-v3[2]模型+迁移学习的方式，进行快速类别扩充。GoogLeNet 在扩大网络提升效果的同时拥有更好的计算效率，它是一个庞大的网络结构，如图 6。

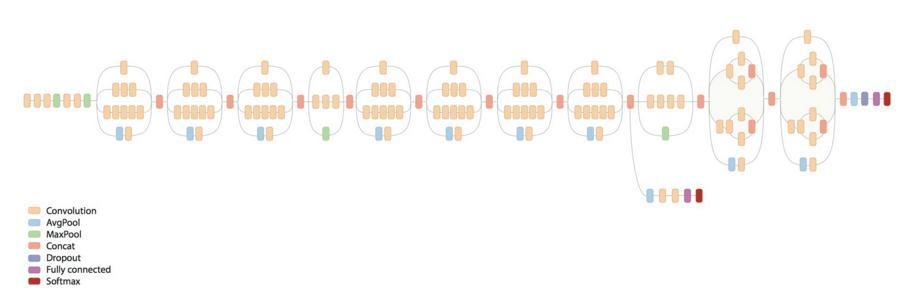


图 6 GoogLeNet 神经网络结构

其中 inception 模块是其核心，具体 inception 结构如图 7。

40

人工智能篇

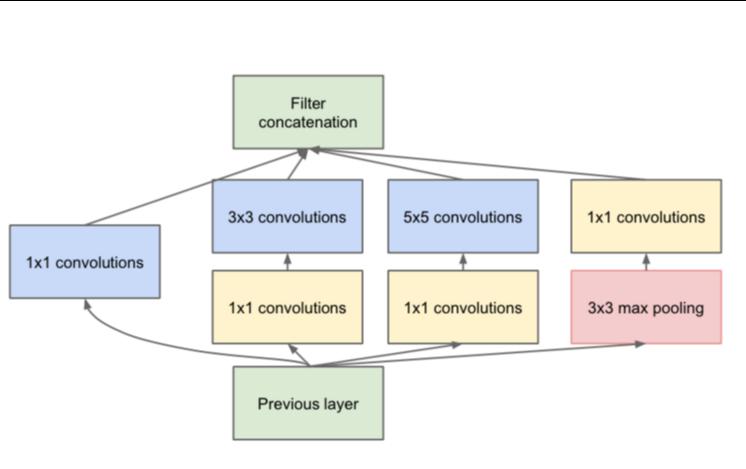


图 7 Inception 神经网络结构

使用基于 ImageNet 的 Inception-v3 模型，由于我们多分类数据量比较小，同时数据内容 ImageNet 内容有一定的差异性，因此并没有直接利用网络的高层进行特征的抽取，而是打开高层网络和浅层分类网络同时进行训练，结果表明，该方法比直接利用网络高层进行特征提取效果提升了 8 个百分点。这是由于网络的高层是对特征的进一步抽象和提取，跟原任务数据具有较高的耦合性，而底层网络抽取的特征则为一些基础特征，在此基础上融合分类网络的训练，对新任务具有更好的拟合性。最终在旅游数据集上，我们在内部的类别标签体系上取得了 92.5%的 mAP。

在此同时，相似类别语料的添加也显得尤为重要，譬如对于“温泉”类别特别容易识别成“小河”或者“游泳池”，添加相近的类别语料供算法进行学习，能够有效提升识别结果的精确度。如图 8，列出了图片的识别结果。

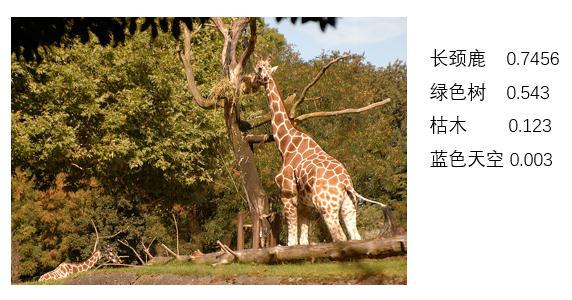


图 8 图片识别结果

41

人工智能篇



**四、成诗引擎**

传统诗歌生成主要借助统计模型，同时结合人工规则。在统计机器翻译的方法中[4]，使用统计机器翻译模型结合韵律规则进行下一句的生成。在遗传算法中[3]，则根据开发者对诗歌的理解，人工定义韵律，流畅性，主题相关性等各个评价因子，构建评估函数，同时使用统计语言模型进行诗歌的自动生成。

随着深度学习技术的日益发展，使用神经网络进行诗歌自动生成是现在的流行趋势。传统的诗歌自动生成模型，由于古诗语言表达的简洁性和语料数量的限制，基于统计的语言模型具有比较严重的稀疏性。在深度学习中，利用 RNN 训练和获取诗歌的语言模型，在一定程度上缓解了该问题[5]。

在此同时，encoder-decoder 框架做为成诗的基本模型，利用 encoder 模块进行历史内容[6][8]、主题内容[7][8]的编码，decoder 模块进行诗句的生成，对主题关联性和语义连贯性，使用不同方法进行尝试，从而构建相关模型。模型[8][9]根据给定的关键词首先对每句话进行主题的规划，从而避免在逐句成诗过程中，主题关联性的弱化。模型[8][6]利用 attention 机制，在成诗过程中进行主题、历史生成内容的融合，而模型[7]则是使用 hierarchical 的 RNN 框架，进一步保证整体主题和语义的统一性。

对于小诗机而言，需要结合多重元素空间和多维主题，而且诗歌语料和主题体系相关性存在缺失，比较难进行端到端的深度学习模型的应用，因此结合了传统模型和深度学习模型，具体从如下几个维度进行了综合优化：

主题规划和相关性：

* 对景点/图片与主题元素的相关性进行打分，多个维度进行综合排序，使整个主题元素空间在全局上权重化，同时结合一定的写诗惯例，基于概率模型动态预先确定每句话的主题元素。
* 基于关键词语义空间的相关主题词获取。

语义流畅性：

* 利用 RNNLM[21]在一定程度上缓解传统统计语言模型的稀疏性，提升诗歌的流畅性，如图 9。

42

人工智能篇

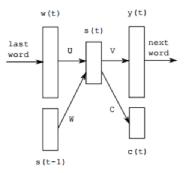


图 9 RNN based language model with classes

* 基于语义层级体系获取语言模型，从多个层次上来把握诗歌的语义搭配粒度， 使得表达更加的流畅。

选优框架

* 基于贪婪算法和局部最优的二排算法进行诗歌的自动生成，从而在保证性能和诗歌基本韵律的基础上，一定程度上确保诗歌主题的相关性和表达的多样性。
* 遗传算法也是对解空间寻求最优化，把古诗的生成看作是一个状态空间搜索问题。将主题相关性，诗歌流畅度、韵律要求等综合融入到评估函数中，适应度函数的设计则较为复杂，同时遗传算法速度较慢，更适合对于速度要求不高的场景。

**五、总结**

小诗机在总体上，是从技术和产品维度上的创新和探索，一方面构建了大规模全面的旅游知识图谱，另外一方面结合深度学习的方法进行基于图片的成诗，使得诗具有较好的主题性和流畅度。

在此基础上，后续会进行进一步的标签细化，包括知识库和图片识别两个方面，使得小诗机拥有更加宽阔的旅游视野和更加细腻的观察维度。在此同时，进行成诗引擎的深入优化，适应更加多元化的元素，真正引领智慧旅游。

**【相关文献】**

1. http://cs231n.github.io/transfer-learning/
2. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[J]. 2015:2818-2826.
3. 周昌乐, 游维, 丁晓君. 一种宋词自动生成的遗传算法及其机器实现[J]. 软件学报, 2010, 21(3):427-437.
4. He J, Zhou M, Jiang L. Generating chinese classical poems with statistical machine

43

人工智能篇



translation models[C]. Twenty-Sixth AAAI Conference on Artificial Intelligence. AAAI Press, 2012:1650-1656.

1. Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]. INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September. DBLP, 2010:1045-1048.
2. Wang Q, Luo T, Wang D, et al. Chinese song iambics generation with neural attention-based model[C]. International Joint Conference on Artificial Intelligence. AAAI Press, 2016:2943-2949.
3. Rui Y. i, Poet: Automatic Poetry Composition through Recurrent Neural Networks with Iterative Polishing Schema. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. IJCAI-16,2238-2244
4. Wang Z, He W, Wu H, et al. Chinese Poetry Generation with Planning based Neural Network[J]. 2016.
5. Ghazvininejad M, Shi X, Choi Y, et al. Generating Topical Poetry[C]. Conference on Empirical Methods in Natural Language Processing. 2016:1183-1191.
6. https://zhuanlan.zhihu.com/p/25084737?columnSlug=easyml
7. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2015:1-9.
8. Yann LeCun, Leon B, Youshua B, Patrick H. Gradient based Learning Applied to Document Recognition. PROC OF THE IEEE, NOVEMBER 1988
9. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
10. Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.
11. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-

778.

[16]何炎祥, 罗楚威, 胡彬尧. 基于 CRF 和规则相结合的地理命名实体识别方法[J]. 计算机应用与软件, 2015, 32(1):179-185.

[17]Wu W, Zhang B, Ostendorf M. Automatic Generation of Personalized Annotation Tags for Twitter Users.Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA. DBLP, 2010:689-692.

1. Otero-Cerdeira L, Rodríguez-Martínez F J,Gómez-Rodríguez A. Ontology Matching: A

Literature Review [J]. Expert Systems with Applications, 2015, 42(2):949–971.

1. Lambrix P, Tan H. SAMBO—A System for Aligning and Merging Biomedical Ontologies [J]. Web Semantics Science Services and Agents on the World Wide Web, 2006, 4(3):196-206.
2. Li J, Wang Z, Zhang X, et al. Large Scale Instance Matching via Multiple Indexes and Candidate Selection[J]. Knowledge-Based Systems, 2013, 50(3):112-120.
3. T Mikolov , S Kombrink , A Deoras ,L Burget , Č ernocký. RNNLM --- Recurrent Neural Network Language Modeling Toolkit.

44

人工智能篇

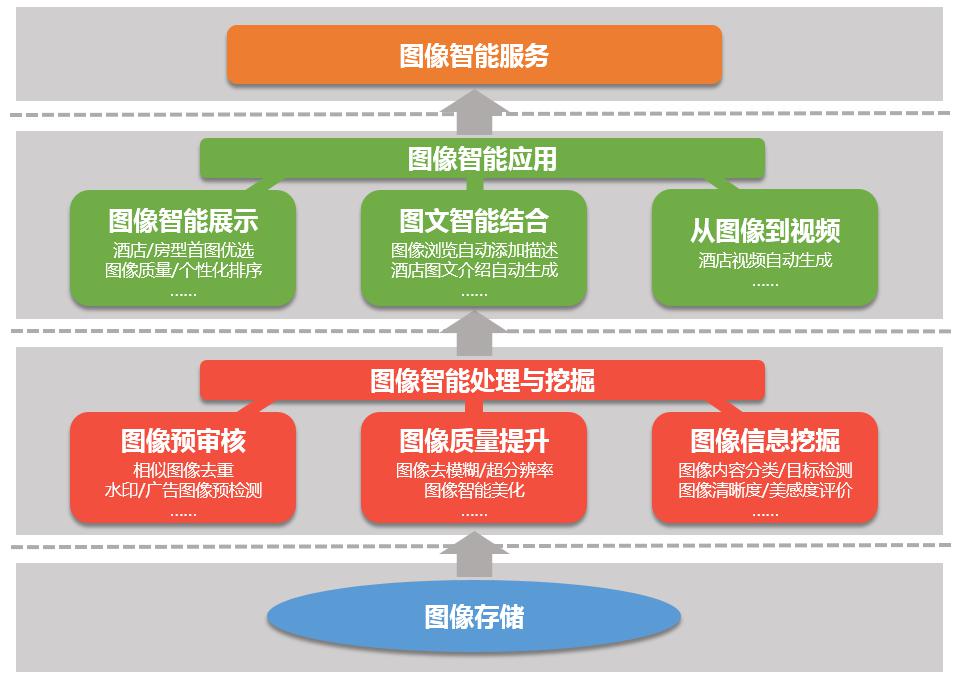


**携程图像智能化建设之路**

**[作者简介]**李翔，携程数据智能部信息科学组图像技术负责人，专注于计算机视觉和机器学习的研究和应用，现阶段致力于酒店图像智能化，在包括 ICCV 和 CVPR 在内的学术会议和国际期刊上发表 10 余篇论文。

携程作为 OTA 行业的领跑者，拥有全球百万家酒店数以亿计的酒店图像，酒店图像数量还在以每天数十万的速度增长。面对海量酒店图像，如何完成智能处理与挖掘，大幅减少图像的人工干预，又如何实现智能应用，改善用户获取酒店信息的速度、准确性和完整性，提高用户满意度，这些都成为急需解决的问题。

相比学术界追求的模型创新性，我们更加关注技术实践在落地场景的效果，力求以简单而有效的方法来解决实际业务问题。为解决上述问题，我们从 0 到 1，围绕酒店图像智能化进行了一系列研究和探索。当前酒店图像智能化的整体架构如下图所示。



其中，通过图像智能处理与挖掘，能够节省大量人工处理图像的成本，挖掘丰富的图像信息，在此基础上，通过图像智能应用可以进一步有效地为用户和酒店创造巨大价值。下面我们将围绕这两个部分展开，从多个具体实践出发，分享携程酒店图像智能化从 0 到 1 的建设之路。

**一、图像智能处理与挖掘**

45

人工智能篇



图像智能处理与挖掘是酒店图像智能化的基础，包括图像预审核、图像质量提升和图像信息挖掘三个环节。

**1.1 图像预审核**

图像预审核是整个酒店图像智能化的第一步，旨在通过一系列图像技术，辅助人工高效地完成海量图像的审核工作，减少人力成本的投入，当前包括相似图像自动去重、水印/广告等视觉可见的不合规图像的预检测等等。下面介绍其中一些实践。

相似图像去重

酒店图像之间的相同/相似主要表现为 1）尺寸形变；2）裁剪残缺；3）色彩变化；4）旋转变化；5）拍摄视角移动等多种情况，如下图所示。



相似图像去重一般分为 1）图像特征表达的提取和 2）图像之间相似度计算两个主要步骤。对于图像特征表达的提取，常见的手工设计特征有颜色、纹理、HOG、SIFT 和 SURF 等；此外基于深度学习的深层特征表达也经常被使用。对于图像之间相似度计算，常见的无监督距离度量方法有欧式距离、曼哈顿距离和余弦距离等；常见的有监督距离度量方法有 LMNN、KISSME、LFDA 和 MFA 等。然而这些方法基于浮点特征计算相似度，计算速度普遍较慢，因此通过哈希学习方法将图像特征转换为二元编码，再利用汉明距离进行相似度的快速计算更加符合工业界对图像数据处理速度的要求。

对于相同/相似的酒店图像，大部分全局特征（比如颜色、纹理和 HOG）不能很好地解决图像裁剪残缺和旋转变化等问题；一些局部特征（比如 SIFT 和 SURF）与基于深度学习的特征虽然表达效果较好，但是由于特征提取复杂，计算速度过于缓慢。

针对以上特征提取方法存在的缺陷，我们采用一种快速特征点提取和描述算法 ORB 作为图

46

人工智能篇



像的特征表达，并使用汉明距离完成相似度计算。ORB 特征具有以下优点：1）特征提取速度快；2）在大多数情况下，去重效果能够与 SIFT/SURF 持平；3）提取的特征直接是二元编码形式，无需使用哈希学习方法就可以直接利用汉明距离快速计算相似度。

在真实应用中，我们还对其进一步优化，从而弥补 ORB 特征不具有尺度不变性的不足，也降低了形变和模糊等因素对 ORB 特征的影响，在保证性能的同时，提升了图像去重的准确率。

图像水印检测

水印在图像中的视觉显著性很低，具有面积小，颜色浅，透明度高等特点，带水印图像与未带水印图像之间的差异往往很小，区分度较低。

我们将图像水印检测问题转化为一种特殊的单目标检测任务。在深度学习兴起前，可变形部件模型（DPM）一直是流行的目标检测方法。深度学习出现后，以 R-CNN、SPPNet、FastR-CNN、FasterR-CNN、SSD、YOLO/YOLO2 等为代表的一系列基于深度卷积神经网络的目标检测方法成为主流。

一个鲁棒的目标检测网络，需要大量的标签数据来支撑，然而收集和标记一个大规模水印目标检测数据集是一件十分耗时耗力的工作。为了解决该问题，我们设计了一套训练数据自动生成和自动标记的数据集制作系统，以较小的人力投入建立了一个多元化的大规模水印目标检测数据集。

在该数据集上，我们进一步对比 FasterR-CNN、SSD 和 YOLO2 三种主流的目标检测方法。通过对性能和效果的综合评估，我们选择在 YOLO2 的基础上进行改进，使其更加适应水印单目标检测任务，从而实现最终的水印图像检测器。基于该检测器，我们实现对数据集中常见上百种水印的完美检测，对数据集中未出现的水印也展现出很好的检测效果。

**1.2 图像质量提升**

图像质量提升旨在通过一系列图像技术，改善酒店图像的质量，进一步减少人工对图像的处理成本，当前包括图像去模糊、小图放大和图像智能美化等等。下面主要分享我们在酒店小图放大中的一些实践。

酒店小图放大

酒店如果存在低分辨率图像通常会被转为高分辨率图像展示给用户，从而使用户获取更多图像细节，更好地增加对浏览酒店的了解。然而，低分辨率图像直接放大，图像会显得模糊，图像细节信息很难恢复。

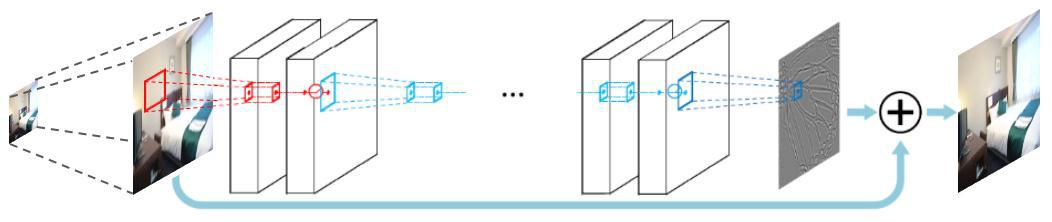
为了令低分辨率图像放大后变得更加清晰，我们引入图像超分辨率技术。最简单的图像超分辨率方法是图像插值，但插值后的图像很容易形成锯齿边缘，细节恢复效果较差。传统的图像超分辨率方法一般采用稀疏表示和字典学习的方式来实现，利用大量高-低分辨率样本对作为先验信息进行图像细节恢复，常见的方法有 SR、ANR、SF 和 A+等。随着深度学习的发

47

人工智能篇



展，通过全卷积神经网络学习低分辨率图像到高分辨率图像的端到端映射函数成为主流方法，其中以 SRCNN、DRCN、VDSR、SRResNet、SRGAN 和 SRDenseNet 等方法为典型代表。



在真实的小图放大场景中，我们选择 VDSR 来构建超分辨率网络，结构如上图所示。具体地，我们组合多种倍数的图像样本进行混合训练，令一个模型能够同时适应多种不同倍数的超分辨率。

然而，在实际应用中我们还遇到一些问题：1）超分辨率网络使用的损失函数一般是最小均方误差（MSE），该函数使重建结果有较高的信噪比，但是缺少高频信息，会使图像出现过度平滑的纹理。2）真实的低分辨率酒店图像往往存在有损压缩，图像本身具有块效应，直接使用超分辨率网络恢复细节，会使图像的块效应更加严重。

为了进一步提升图像超分辨率的效果，我们在 VDSR 基础上，针对上述问题做了一系列改进，在保证网络输出的图像更加自然的同时，也大幅降低了块效应的影响。利用上述模型，低分辨率图像的高分辨率重建能够被高效地实现，酒店图像质量得到明显提升，人工处理图像的成本大幅减少。

**1.3 图像信息挖掘**

图像信息挖掘旨在通过一系列图像技术，自动快速地挖掘出图像中蕴含的丰富内容，为每张酒店图像都建立一个信息档案，为下一步图像智能应用打下坚实的基础，当前包括图像内容分类、图像多目标检测和图像质量评价等等。下面对其中的一些实践做简单介绍。

图像内容分类

酒店图像是对酒店各方面信息的直观展示，因为我们需要尽可能地帮助用户方便快捷地发现他们想要浏览的图像内容，所以酒店图像的内容分类显得尤为重要。

随着深度学习的出现，尤其是卷积神经网络的兴起，利用大量已经标注类别的酒店图像样本直接训练一个深度卷积神经网络，比如常用的 AlexNet、VGGNet、ResNet、DenseNet 和基于 Inception 的一系列网络等，就可以实现对酒店图像的分类。然而，如果通过人工标注大量训练样本代价极大，如果训练样本不足则会导致网络过拟合。为了能够实现在标注少量酒店图像的情况下达到良好的分类效果，我们利用深度网络有效的迁移学习能力，对在大规模数据集上已经预训练的网络权重进行微调。

在实际应用中，我们没有借助应用最为广泛的 ImageNet 数据集，因为该数据集图像的内容和酒店图像差异过大，影响了网络迁移学习的效果。为了尽可能的提升网络的迁移能力，我

48

人工智能篇



们借助与酒店图像内容最为接近的自然场景图像数据集上预训练的 VGGNet 作为初始设置。同时，我们进一步利用水平翻转、随机裁剪和色彩抖动等方式对标注的小规模酒店图像数据集进行数据增强。

基于以上的方法，我们实现了酒店图像内容十余个类别的准确区分，为之后图像智能应用做好了准备。

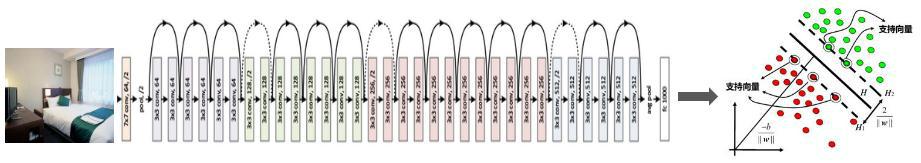
图像质量评价

在上一节中，我们介绍了通过酒店图像分类模型来挖掘图像的类别信息。接下来我们需要更进一步对所有酒店图像进行质量评价，为每张酒店图像计算质量分数来表征其质量的高低。

起初我们选择清晰度这一客观指标作为图像质量评价的标准，然而我们发现仅以清晰度作为图像质量评价的标准存在不足，清晰度高但内容不好看的图像为数不少。因此我们更希望能够从美学角度来对图像质量进行评价。

图像美感度是一个非常主观的概念，很难有一个统一的标准去量化，为了能够尽可能准确地量化图像的美感度分数，我们选择深度学习的方法来实现美感度评价。具体地，我们把计算分数的回归问题转化为判断图像好看与否的分类问题，为了把模型的判断结果转化为美感度分数，我们在模型的最后一层输出图像被判断为好看的概率，作为图像的美感度分数。通过这种方法可以很直观地把一个二元分类器的输出结果转化成打分的结果。

在实际应用中，我们又再次遇到了同样的问题，缺少大量标注了好看/不好看标签的训练图像。由于在酒店图像分类中，我们利用卷积神经网络强大的迁移学习能力取得了成功，所以我们决定继续沿用这种方法。由于酒店图像的美观度受到内容、色彩和构图等多方面的影响，所以我们不再像酒店图像分类那样只借助内容单一的场景图像数据集，而是将包罗万象的 ImageNet 数据集和场景图像数据集混合进行 ResNet 模型的预训练，力求让尽可能多的图像参与层数更多的 ResNet 模型的学习，令更深的网络能够记住更多图像的内容，对图像内不同区域之间关系的理解更加深入，从而进一步提高网络在美感度评价上的迁移能力。



在数据集标注的过程中，由于图像好看与否主观性依赖很强，所以我们综合了多人评判结果作为每张酒店图像好看与否的标注。由于图像经过翻转、裁剪或色彩抖动都会令图像的美感度发生变化，所以我们未对数据集进行数据增强。为了防止直接微调网络出现过拟合，我们转而利用特征迁移，在 ResNet 的深层特征表达基础上，训练支持向量机实现酒店图像好看/不好看的二元分类模型，模型结构如上图所示。通过图像质量评价，我们获取了酒店图像的质量分数，为后续图像智能应用提供了重要依据。

**二、图像智能应用**

49

人工智能篇

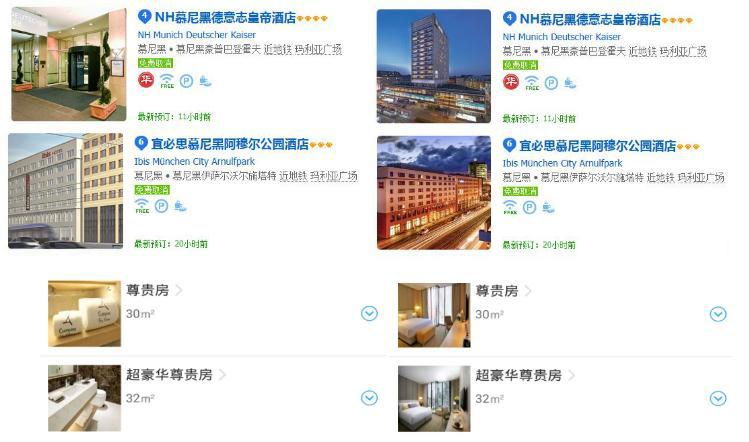


图像智能应用能够为用户和酒店创造巨大价值，是图像智能化的重要组成部分，当前包括图像智能展示、图文智能结合和酒店视频等应用场景。

**2.1 图像智能展示**

酒店和房型的首图如何挑选才能提升用户的满意度，酒店图像如何排序才能使用户快速获取想要的酒店信息。我们针对这些问题，对图像进行一系列智能展示，在包括酒店/房型的首图优选、酒店图像分类展示和质量/个性化排序等方面做了诸多尝试，努力提升用户体验。

酒店/房型的首图是用户对酒店/房型的初始印象，除了商家或运营人工指定首图外，绝大部分酒店和房型的首图都是由机器进行挑选。为此，我们基于图像智能挖掘获取的图像分辨率、内容类型、清晰度/美感度分数，设计了完整的首图优选模型，力求把图像多种不同信息自适应融合起来，为酒店挑选最优的酒店图像展示给用户。



通过我们的首图优选模型，酒店和房型的首图质量大大提高，对比效果如上图所示（左为原版，右为新版）。更优质的酒店和房型首图使得用户预订转化率显著上升，在提高用户满意度的同时，也给酒店带来了价值。

**2.2 图文智能结合**

在图像单一展示的基础上，我们还进行了一些图像和文本智能结合的探索，包括图像自动添加描述和酒店图文介绍自动生成等工作。

我们尝试对酒店图像自动添加描述，使用户在浏览图像的过程中能够浏览文字，加深其对图像内容的了解。然而，利用基于深度学习的 ImageCaption 模型从图像得到的描述性文字显得生硬而平淡，直接展示给用户很不自然。我们在此基础上进行改进，进一步结合海量用户

50

人工智能篇



点评数据，利用真实用户的评论内容来描述酒店图像。

点评不仅流畅而且自带情感，用户在浏览图像的同时可以看到其他用户对该图像内容的真实评价，方便快捷地提升了用户获取酒店信息的丰富度和可靠性，用户浏览费力度也得到显著下降。下图展示了图像浏览中结合点评描述的应用场景。



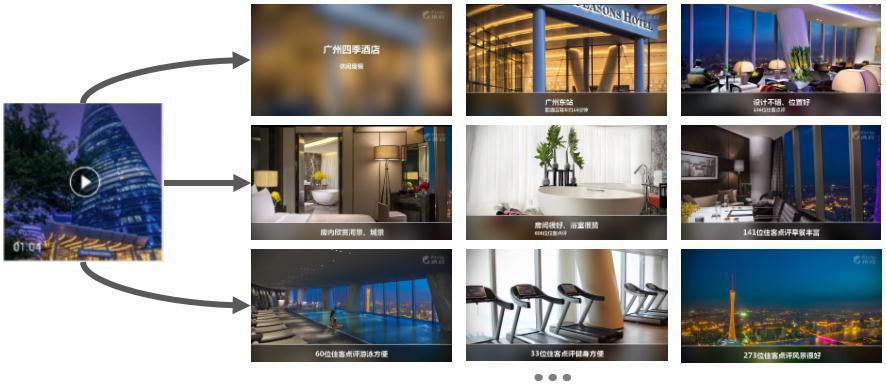
**2.3 从图像到视频**

随着移动通信的发展和 WIFI 的普及，用户已经不再满足于只浏览静态图像，视频的观看需求日益增长。我们在这方面也做了一些实践，通过视频播放形式对酒店信息进行连续展示，使用户观看酒店视频就可以对酒店信息进行更全面的了解，减少用户的浏览费力度，提升用户体验。

为每家酒店拍摄视频需要投入大量人力物力，为了快速获取酒店高质量精美视频，我们利用酒店图像来自动生成酒店视频。具体地，我们借助图像挖掘中获取的丰富的图像信息，设计了一个智能选图系统，实现视频图像的自动挑选。同时，我们通过酒店的文本信息自动匹配相应的字幕。下图展示了一个酒店视频包含的部分图像和字幕。

51

人工智能篇



酒店视频上线后，取得了很好的反响，用户通过观看酒店视频加深了对酒店的了解。现在酒店视频日均观看数万次，用户预订转化率和间夜量显著上升，用户和酒店实现了双赢。我们还在继续对酒店视频不断优化和迭代，力求为用户和酒店创造出更大价值。

**三、总结与展望**

我们通过介绍携程多个真实的图像智能化案例，分享了从 0 到 1 的图像智能化建设之路，但计算机视觉和机器学习对于携程图像智能化的价值远远不限于此。接下来我们将继续在多个图像应用场景进行深入挖掘，力求为携程图像智能化贡献更多的力量。

52

人工智能篇



**面向前端工程师的机器学习引导课**

[作者简介]古映杰，携程度假研发部高级研发经理。负责前端框架和基础设施的设计、研发与维护。开源库 react-lite 作者。

本文面向前端工程师，使用熟悉的前端技术 JavaScript，讲解机器学习里的基础概念和算法。

演示可在浏览器里运行的机器学习 DEMO，包括拟合线段中心点，拟合矩形中心点，线性回归，AI 玩 Flappy-Bird 和 2048 游戏，识别手写数字等效果。

**一、什么是人工智能（AI）？**

智能行为和现象，有不同的来源。有的来自生物，有的来自机器。

我们可以把那些来生物的智能，称之为自然智能。它们是通过自然选择逐步演化而来的智能。

而另外一些由人类设计的机器所表现的智能，就是人工智能，简称 AI。

**二、人工智能的两大分类**

按照解决问题的能力，我们可以把人工智能，分成两类：

强人工智能：拥有自我意识，具备解决通用问题的能力弱人工智能：没有自我意识，具备解决特定问题的能力

目前，我们能看到的人工智能，几乎都是弱人工智能，在解决特定问题的能力上，超越了人类。

**2.1、强智能之拉普拉斯妖**

我们可以通过物理学里的一个思想实验，来侧面理解强智能，是怎样的。

拉普拉斯是著名的数学家，他提出了一个看法：如果一个智能体，知道宇宙中每个原子确切的位置和动量，那么就可以通过牛顿定律推演出宇宙的过去以及未来；宇宙中的一切问题，都可以得到精确的解答。

53

人工智能篇

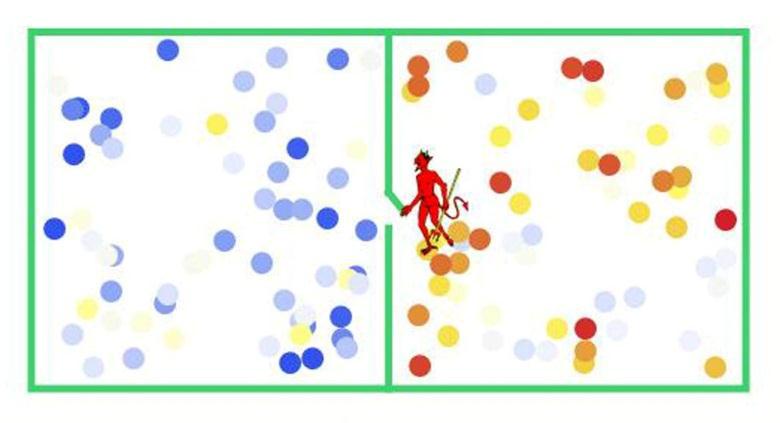


然拉普拉斯妖，后来被证明是不可能的。但它确实反映了解决通用问题的一种做法。

**2.2、弱智能之麦克斯韦妖**

另一个物理学的思想实验，来自麦克斯韦。这个思想实验的目的，是为了挑战热力学第二定律。该定律指出，封闭系统最终会达到热平衡。

于是，麦克斯韦假设存在一种智能体：麦克斯韦妖，看守暗门，观察分子运动速度。使较快的向某侧流动，较慢向另一侧流动。经过充分长的时间，两侧温差会越来越大 。温度高低是分子运动剧烈程度的宏观表现，通过分隔不同运动速率的分子，就让系统的两个部分的分子有了不同的剧烈程度。



**三、什么是机器学习？**

机器学习，是英文 Machine Learning 的直译。它是实现人工智能的其中一种方式。前面说过的拉普拉斯妖就是不是机器学习的方式。

人类的学习，可以归纳为这种现象：随着经验的增加，解决问题的能力得到提升。

对机器而言，经验其实就是数据。机器学习就是：用数据训练程序以优化其表现的算法。

机器学习是一个相对宽泛的概念，只要满足它的定义，就属于机器学习。并不是一定要 GPU/TPU 训练，一定要多少行代码，一定要解决多宏大的难题，才叫机器学习。用数据训练一个模型，拟合一个点，也是机器学习的体现。

54

人工智能篇



**四、机器学习的分类**

机器学习有两种分类：监督学习和非监督学习。

区分这两种学习方式的依据很简单，就是训练数据是否包含了答案。包含答案，就是监督学习；不包含答案，就是非监督学习。当然，这里也存在中间状态，包含一个渐进式答案，或者其它形式的间接答案等，就叫半监督学习。

**五、什么是深度学习？**

深度学习，是英文 Deep Learning 的直译。它是实现机器学习的其中一种方式。机器学习还包含其它实现方案。

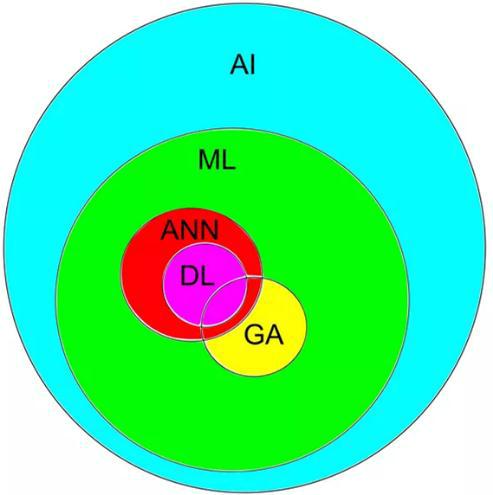
深度学习里，用到了人工神经网络，这是一个用计算机模拟大脑神经元运作模式的算法。同时，这个人工神经网络的隐藏层数量还必须足够多，才能构成深度神经网络。然后喂之以大量的训练数据，就是深度学习了。

换一个角度，如果隐藏层数量不多，而是每个隐藏层里包含的神经元数量很多，在形态上，它就是一个往宽度发展的神经网络结构。这时，可能就叫广度学习了。

目前，深度学习还是主流，它的训练效率，优于广度学习。

**六、AI & ML & DL & GA 的关系**

下图展示了人工智能，机器学习，人工神经网络，深度学习和遗传算法之间的关系。我们可以看到，除了遗传算法是交集关系意外，其余的是分离出一个个子集的关系。



**七、前端工程师与人工智能**

前端工程师跟人工智能，有什么关系呢?

55

人工智能篇



这个问题应该反过来问，首先，按照目前的发展，将来人工智能会跟所有人产生紧密关联。前端工程师也是人类，作为人类，应该在某种程度上了解人工智能，而后能更好地使用人工智能的产品。

其次，前端工程师，也是程序员。我们可以在程序员的层面，比普通人更好地理解人工智能背后的机制。

再次，我们才是前端工程师。在前端工程师的角度，去审视人工智能未来是否会取代我们的工作。它取代我们之后，会不会有新的工作岗位出来，比如 AI 前端工程师。这时，那些对人工智能更加了解的前端工程师，就更容易得到相关的岗位了。

更何况，了解人工智能，说不定有机会转型成 AI 程序员。

编程语言、开源社区、IDE 等持续发展，会不断降低人工智能的开发门槛。过去的经验告诉我们，五年前高级程序员才能做到的事情，现在一个的普通程序员也有望做到。

人工智能这个概念，是在上世纪 50~60 年代提出的。当时，关心人工智能的那帮人，都是计算机里的鼻祖人物和数学家们。但是现在，作为前端工程师的我，居然也能写上一些代码，在 web 页面上跑起机器学习的 DEMO，这正是反映了门槛的降低趋势。

**八、UI 开发的三种模式**

手写标签和样式代码，生成页面

可视化拖拽 UI 组建，生成页面

直接输入设计稿，输出可用页面(https://github.com/tonybeltramelli/pix2code)

要想在前端自动化上做到极致，也无法跳开人工智能环节。pix2code 这个项目就是一个案例。虽然现在它的能力还很弱，使用场景很有限。但我们不要忘记，人工智能的发展速度，可能是指数级的。谷歌的阿尔法狗，一开始只是打败了人类业余围棋玩家，后面战胜了人类顶尖玩家李世石，再后面击败了当时的世界第一柯洁。然后，再无对手，又开始抛弃人类经验，通过自我对弈的方式，打败之前的版本。再到后面，竟已不局限于围棋，拓展到了国际象棋，日本将棋等其它棋类游戏里。这里的发展速度，是非常快的。

所以，不能因为 pix2code 目前的能力，而盲目乐观。

**九、数学知识回顾：什么是导数？**

霍金在写《时间简史》时说过，书里每出现一个公式，书的销量就减半。这反映了公式在科普和推广阶段的负面作用。

但是，不用公式和代码，又要讲明白机器学习，几乎是不可能的。听着玩儿的话，我们提供的 DEMO 和前面的概念介绍，也足够满足要求了。

接下来，是直面公式和代码的时刻，不能逃避，在真正理解后，你会发觉，原来公式才是最

56

人工智能篇



简单、最容易理解的那个。那些比喻、类比和段子，最后都不会在头脑里保留多久，只是在短时间内，营造了学到东西的虚假体验。

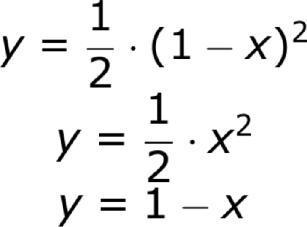


上面是导数的代数形式，下面是导数的几何形式。我们可以看到，导数，其实是围绕一个点来谈论的。当我们说一条线的导数是多少多少时，只是一个特例，恰好那条线上的每个点的导数都是同一个值。

当我们选取了一个点 x0 之后，在 X 轴上追加一个无穷小的增量，然后用 X 加上无穷小增量得到 x1，求得 x1 对应的 Y 轴的值 y1 之后，通过 x1 - x0 和 y1 - y0，我们得到了一个小三角形。这个小三角形的 y 轴长度，除以 x 轴长度，就时这个点 x0 的导数值。

关键点有两个。一个是找到小三角形，它是直角三角形。第二个除法，是用两个直角边相除，就得到了导数。导数反映了这个点跟下一个点的变化幅度和趋势。

**十、数学知识回顾：什么是复合函数？**

****

不是所有函数都那么简单，很多函数很复杂，甚至不能写在一条公式内，甚至不能用公式表达出来。

不过，我们还是可以从函数组合的角度，对许多复杂函数进行解构。

比如上面的第一条函数，虽然它本身已经够简单了。但其实还可以拆分成两个子函数的组合。

先计算第三条公式的结果 Y，在把 Y 作为 X 值，带入第二条公式内，求得另一个 Y 值。

57

人工智能篇

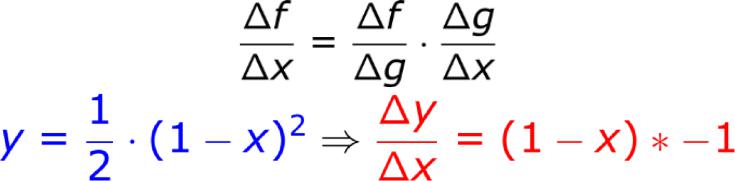


这个过程就相当于对第一条公式进行 Y 值计算。

**十一、数学知识回顾：什么是链式法则？**

简单函数，有一些导数计算公式可以套用。那复杂函数里，又应该如何求导呢？

前面我们了解了复合函数的概念，它可以拆分成简单函数之间的组合关系。通过这个组合关系，我们可以用简单函数的导数，计算出复杂函数的导数。



在介绍导数时，我们特别强调了“除法”，链式法则就是利用，除法和函数组合时产生的分子和分母的颠倒关系，不断地链式相乘，消掉中间变量，最后得到我们想要的目标导数值。

**十二、数学知识回顾：什么是梯度下降？**

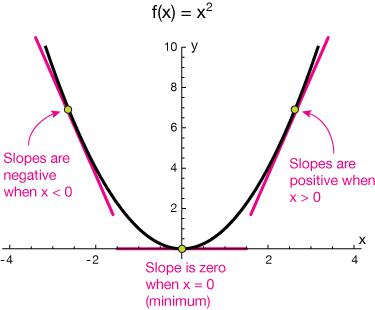
前面的数学知识部分，已经是本次分享里最难的环节了。后面是相对轻松的部分。

梯度下降，是机器学习里的重要概念。不管你去看谁的书或者视频，都绕不开这个概念。可能其中有些老师会打个比方说：假设你在一座山的山顶或者山腰上，你周围弥漫着浑厚的迷雾，无法看清下山的路，这是你要下山，要用什么方式？就是用脚探索更低点，然后逐步走下来。这就是梯度下降。

听了这个比喻，你理解了梯度下降吗?

反正我没有，下山我会，梯度下降嘛，还是不懂。

我们可以看到，比喻终究是间接的，含混的。还是要直面更直接和纯净的知识。



58

人工智能篇



梯度下降，缘于这么个观测事实：

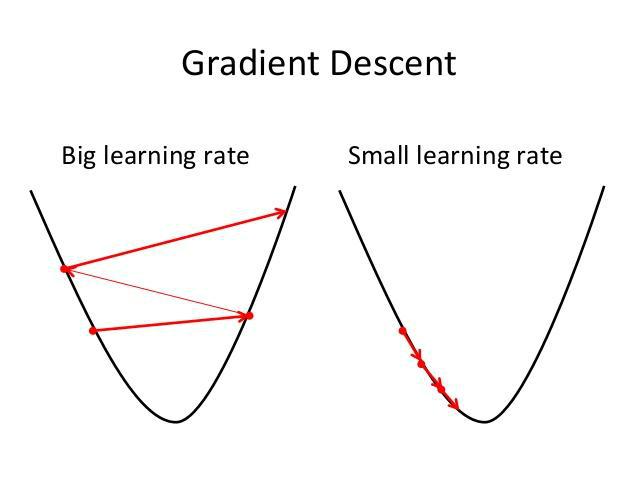
1）当我的导数是一个正数时，更低点在 -X 轴的方向； 2）当我的导数是一个负数时，更低点在 +X 轴的方向； 3）那么，不管我的导数是正数还是负数，它的反方向，就是更低点的方向。 4）沿着导数的反方向走，必将走到一个相对最低点。

这就是梯度下降。如果你理解了它，那你可以很容易地理解，什么是梯度上升。就是更高点，在沿着导数的方向嘛。

上图是一个 U 形图，只有一个最低点。如果是一个波浪图，那就有很多个局部最低点，如何找到全局最低点，目前也没有完美解决方案，还是一个难题。

**十三、数学知识回顾：什么是学习率？**

梯度下降，只是为我们指明了更低点的方向，但却没有告诉我们距离。所以我们必须选择一个步伐大小，这个步伐大小，被称之为学习率。



如果设置的学习率过大，步伐太大，一下子跨到对面去了，下次又走导数的反方向跨回来。

就这样来来回回地振荡，没有走到一个令人满意的低点。

如果设置的学习率过小，走了很久，也仿佛在原地踏步。这也不行。

所以，选择一个合适的学习率，很重要。如何自动选择最优学习率，目前也是一个难题。

**十四、前端人工智障之拟合线段中心点**

在线 DEMO

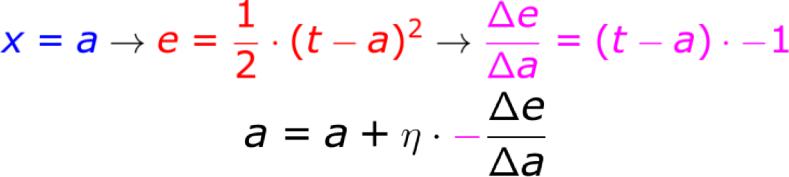
<https://lucifier129.github.io/simple-machine-learning-demo/01/index.html>

源代码

<https://github.com/Lucifier129/simple-machine-learning-demo/blob/master/01/index.js>

59

人工智能篇



要拟合一个点，先要设置一个预测模型，在这里用 x = a 就可以了。最初 a 的值是随机的，或者设置为 0，也可以，权当瞎猜。

然后我们用生成的训练数据来喂给学习算法。数据里都包含了答案，我们就可以得到一个上图误差公式。它叫均方误差。用正确答案减去瞎猜的答案，得到一个值，再对其求平方，保证是正数。

有了均方误差公式，我们就可以用之前学到链式法则求导技巧，把 Y 轴设置成误差值 e，把 X 轴设置成参数 a，然后用梯度下降寻找误差最小的点，下一个 a 参数就是： a + 学习率 \* -1 \* 导数值。其中，（学习率 \* -1 \* 导数值）得到的就是往导数反方向走的那一小步的步长。

当误差最小点为 y = 0 点时，可以说，我们找到了理想的参数 a，它完美的，零误差地拟合了目标点。

值得一提的是，你可以看到误差 e 对模型输出的 y 的导数，由链式法则计算出来，恰好包含了一个 -1，而找最低点，是用导数的反方向，也包含一个 -1，这相乘就是负负得正，可以消掉。在有些机器学习的介绍公式和代码实现里，它们就省略这两个语义不同的 -1。这对于初学者来说，是非常要不得的。它使得梯度下降的形式，看起来像梯度上升，容易混淆概念，徒增学习难度。

**十五、前端人工智障之拟合矩形中心点**

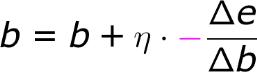
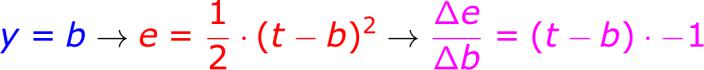
矩形中心点，无非就是上面的 X 轴坐标点的基础上，增加一个 Y 轴的维度嘛。

在线 DEMO

<https://lucifier129.github.io/simple-machine-learning-demo/02/index.html>

源代码

<https://github.com/Lucifier129/simple-machine-learning-demo/blob/master/02/index.js>



每个训练数据，都包含了 (x, y) 两个数据，我们增加一个对 y 轴的参数 y = b。然后用相同的套路，分别同步训练即可。

60

人工智能篇



a = a + learningRate \* - gradientA

b = b + learningRate \* - gradientB

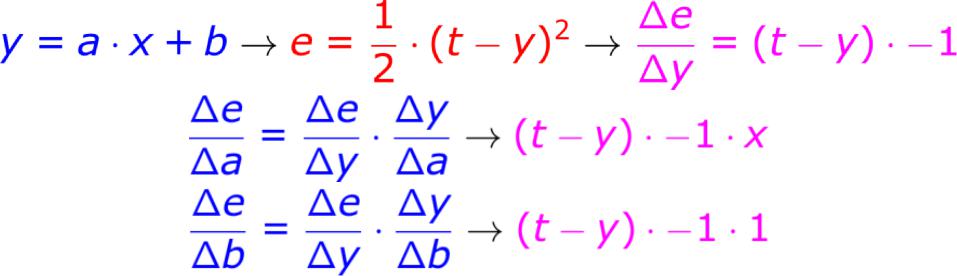
**十六、前端人工智障之线性回归**

在线 DEMO

<https://lucifier129.github.io/simple-machine-learning-demo/03/index.html>

源代码

<https://github.com/Lucifier129/simple-machine-learning-demo/blob/master/03/index.js>



线性回归也是，除了预测模型的公式变成了 y = a \* x + b，其它套路都一样。

在预测时，a 是常量系数，x 是自变量，y 是因变量。当调整参数 a 时，预测时输入的 x 成了 a 的常量系数了。所以对 y 求导 a，得到的值是 x 。

这里存在一个正向计算预测结果，反向修复参数误差的过程。

**十七、什么是感知机(Perceptrons)？**

在线 DEMO

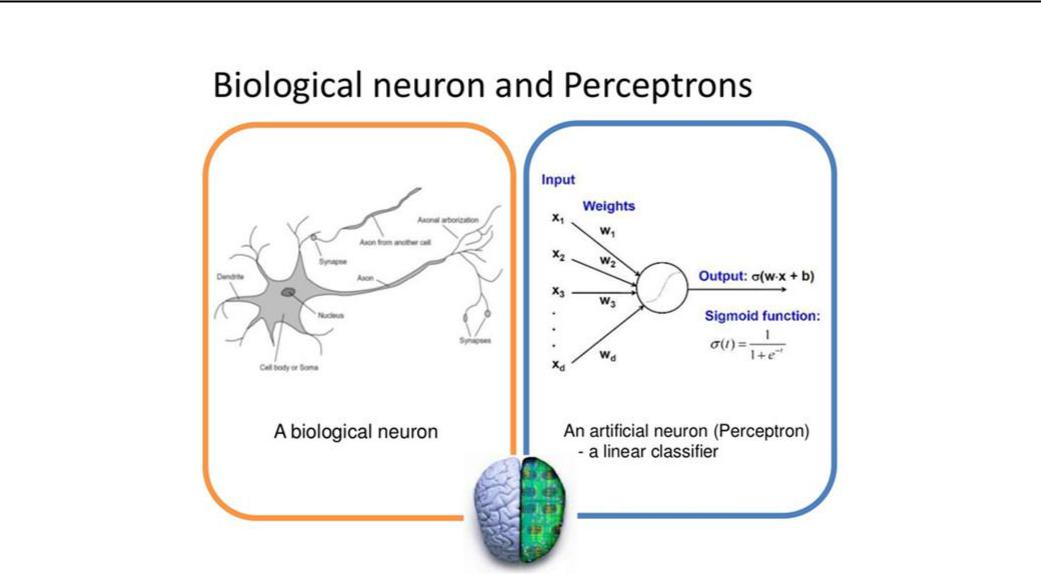
<https://lucifier129.github.io/simple-machine-learning-demo/04/index.html>

源代码

<https://github.com/Lucifier129/simple-machine-learning-demo/blob/master/04/index.js>

61

人工智能篇

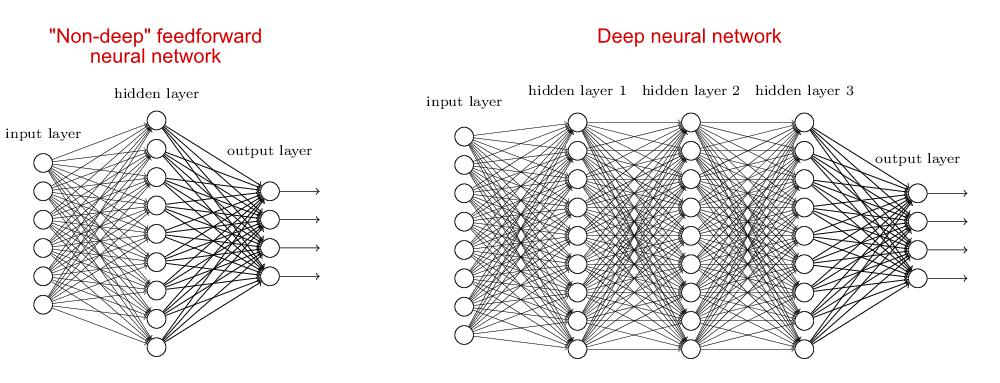


线性回归的公式是 y = a \* x + b，从数学角度，很明显可以推广一下，成为 y = w1 \* x1 + w2 \* x2 + w3 \* x3 + ... + wn \* xn + b 的多因素形式。

在上面的基础上，再增加一个激活函数，就构造了一个被称之为感知机的模式。它最初是模拟大脑神经元在接受到刺激之后，根据是否达到阈值，来决定是否放电，刺激其它神经元。

感知机可以对多因素的事物，进行线性分类。比如对肿瘤进行良性肿瘤和恶性肿瘤的分类，对邮件进行垃圾邮件和非垃圾邮件的分类。不过，感知机的分类实现线性的，如果一个分类无法用线性分割出来，感知机就无法解决它。著名的"与或问题"，就曾经打击到了人工神经网络的研究热度。

**十八、什么是人工神经网络(ANN)？**

****

感知机是线性回归里的公式推广后的形态，而人工神经网络，则可以看成是感知机的推广形态。不只是一个人工神经元，而是多个神经元以某种网络结构链接在一起，配合非线性的激活函数，它可以实现更强大的分类和拟合能力。

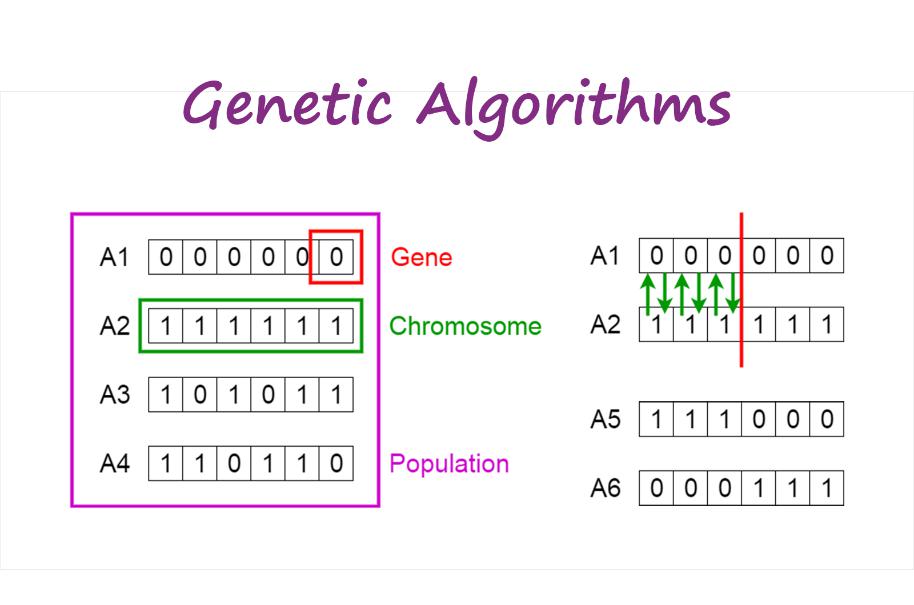
62

人工智能篇



反向传播算法，就是一种对多层人工神经网络进行调参的算法。它的原理，正如我们前面介绍的求导、链式法则和梯度下降。感兴趣的同学，可以尝试自行推导和实现一下反向传播算法。

**十九、什么是遗传算法(GA)？**

****

遗传算法是一个伟大的算法。对人类而言，可能是最伟大的算法。

因为，广义上的算法，不局限于编程领域。任何确定性的、在有限步骤下完成的特定步骤序列，都可以称之为算法。它可以是代码形式的，也可以是物理形式的。

人类本身，就可以看成遗传算法跑在大自然这个生态系统平台上的产物。

现在我们人类，又把遗传算法应用到了计算机平台上。所有能编码到一个序列的问题，都可以用遗传算法解决。区别在于，遗传算法未必是最好最经济的解决方案罢了。

像太阳系，或者宇宙，拥有庞大的资源，可以花费数十亿年的时间，跑遗传算法（自然选择），最后演化出人类智能。我们人类的算力，却着实有限，所以我们通常需要更高性能的算法。

遗传算法模拟了自然选择里的几个概念，基因、染色体以及种群，通过让两个染色体按照某个突变率交换基因位的编码，来不断得到新的，可能更好的解决方案。

遗传算法和人工神经网络结合起来，产生了一个被成为神经进化的算法。后面我们可以看到这个算法应用在玩 Flappy-bird 游戏的效果。

**二十、前端人工智障 DEMO**

神经网络+遗传算法，玩 Flappy-Bird

<https://lucifier129.github.io/factor-network/examples/build/#Flappy-Bird-Of-Neuroevolution-Without-Labeled-Data>

63

人工智能篇



神经网络+反向传播，玩 Flappy-Bird

[https://lucifier129.github.io/factor-network/examples/build/#Flappy-Bird-Of-Back-](https://lucifier129.github.io/factor-network/examples/build/#Flappy-Bird-Of-Back-Propagation)

[Propagation](https://lucifier129.github.io/factor-network/examples/build/#Flappy-Bird-Of-Back-Propagation)

神经网络+遗传算法+变种，玩 Flappy-Bird <https://lucifier129.github.io/factor-network/examples/build/#Flappy-Bird-Of-Back-Propagation>

十大高手，玩 Flappy-Bird

<https://lucifier129.github.io/factor-network/examples/build/#Flappy-Bird-Of-Ten-Masters>

反向传播，识别手写数字

[https://lucifier129.github.io/factor-network/examples/build/#MNIST-Handwritten-Digit-Of-](https://lucifier129.github.io/factor-network/examples/build/#MNIST-Handwritten-Digit-Of-Back-Propagation)

[Back-Propagation](https://lucifier129.github.io/factor-network/examples/build/#MNIST-Handwritten-Digit-Of-Back-Propagation)

Flappy-bird 等游戏，在每一帧里，都没有包含明显的正确决策，所以监督学习在这里比较难以派上用场。我们要用非监督的方式来学习。

第一个 DEMO，遗传算法+神经网络，生成一批神经网络模型，指导每只小鸟是否 flap 飞起来的决策。当鸟儿碰壁死绝，用每只鸟儿得到的 score 分数进行排名，然后让排名靠前的神经网络，繁殖更多，而排名靠后的繁殖更少，或者淘汰。经过 N 代的演化，最后得到优秀的神经网络参数模型。

第二个 DEMO，用第一个 DEMO 训练出来的优秀模型，作为正确答案的生成器，我们就可以得到一个监督学习的用场了。把优秀 Bird 的决策当作正确答案，喂数据给反向传播算法，当两者的误差趋于一个很小的值时，另一个优秀的模型就训练出来了。

第三个 DEMO，还是遗传算法+神经网络，只是这次，不是用分数高低来进行排名。分数不是答案，所以是非监督学习。这次，我们用第一个 DEMO 训练出来的优秀模型，作为正确答案的生成器。然后计算每只演化中的 Bird 模型的误差，误差大，排名靠后，误差小，排名靠前。由于训练包含了正确答案，所以是监督学习。最后，我们演化出了一个跟第一个 DEMO 训练出的优秀模型误差很小的新模型。

如此可见，遗传算法+神经网络拥有更强的解决问题的能力，既可以监督学习，也可以非监督学习。但反向传播算法，只需要调整一个神经网络参数模型，而遗传算法的版本，却动辄成百上千个模型，性能自然不那么好。

第四个 DEMO，是用训练的 10 只优秀 Bird 模型角逐。目前为止，跑到 1 亿分，我也没见它们任何一个倒下。

第五个 DEMO，是机器学习里经典的 MNIST 手写数字训练集，采用的是反向传播算法的监督学习。可以识别你写在画板上的数字。

**二十一、结语**

尽管前端并非机器学习的主场，但作为学习，它可能挺适用的。起码我们可以更容易地在网页上看到效果。

64

人工智能篇



当然，如果想更深入理解机器学习，需要去看更专业的书籍或教程。本次分享，主要是作为一个引导，激发大家对机器学习的兴趣。

65

人工智能篇



**证件全文本 OCR 技术，了解一下**

**[作者简介]**周源，携程技术平台研发中心高级研发经理，从事软件开发10余年。2012年加入携程，先后参与支付、营销、客服、用户中心的设计和研发。

本文从计算机视觉的前世今生，到证件全文本 OCR 的实践，带你了解人工智能、计算机视觉、深度学习、卷积神经网络等技术。无论是计算机视觉的入门者还是从业者，希望都可以有所收获。

**一、什么是 OCR**

光学字符识别（英语：Optical Character Recognition, OCR），是指对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程。

一般的识别过程包括：

图像输入：对于不同的图像格式，有着不同的存储格式，不同的压缩方式，目前有 OpenCV、 CxImage 等开源项目。

预处理：主要包括二值化，噪声去除，倾斜校正等。

二值化：摄像头拍摄的图片，大多数是彩色图像，彩色图像所含信息量巨大，对于图片的内容，我们可以简单的分为前景与背景，为了让计算机更快的、更好地识别文字，我们需要先对彩色图进行处理，使图片只剩下前景信息与背景信息，可以简单的定义前景信息为黑色，背景信息为白色，这就是二值化图。

噪声去除：对于不同的文档，我们对噪声的定义可以不同，根据噪声的特征进行去噪，就叫做噪声去除。

倾斜校正：由于一般用户，在拍照文档时，都比较随意，因此拍照出来的图片不可避免的产生倾斜，这就需要文字识别软件进行校正。

版面分析：将文档图片分段落，分行的过程就叫做版面分析。由于实际文档的多样性和复杂性，目前还没有一个固定的，最优的切割模型。

字符切割：由于拍照条件的限制，经常造成字符粘连，断笔，因此极大限制了识别系统的性能。

字符识别：这一研究已经是很早的事情了，比较早有模板匹配，后来以特征提取为主，由于文字的位移，笔画的粗细，断笔，粘连，旋转等因素的影响，极大影响特征的提取的难度。

版面还原：人们希望识别后的文字，仍然像原文档图片那样排列着，段落不变，位置不变，

66

人工智能篇



顺序不变地输出到 Word 文档、PDF 文档等，这一过程就叫做版面还原。

后处理、校对：根据特定的语言上下文的关系，对识别结果进行校正，就是后处理。

目前的主流实现是 CNN+RNN+CTC 和 CNN+RNN 基于 Attention 的方法。

**二、携程证件 OCR 项目**

**2.1 项目目标**

根据携程的实际使用场景，使用 OCR 技术识别身份证、护照、火车票、签证等证件的中文英文及数字文本信息。

**2.2 主要进展及展望**

2016 年，实现客户端身份证、护照英文数字的识别。

2017 年，实现 Offline 场景身份证、护照、火车票等中文识别。

2018 年，实现 APP 实时识别身份证、护照、火车票、驾驶证、行驶证、签证等中英文文本。

**2.3 大概的精度情况**

2.3.1.数字英文

误识率<0.5%【线上数据统计】

拒识率~5%

干扰因素包括：曝光、倾斜、远照

2.3.2.中文

1:N 有引导（指引导用户将证件放于相机框中）

FAR= 1%【线上数据统计】

拒识率～20%

1:N+1 无引导

FAR= 3%【线上数据统计】

拒识率～30%

-曝光、图像质量低计入 FAR

-遮挡计入拒识

-考虑外籍证件

-考虑少数民族

**三、关键知识**

67

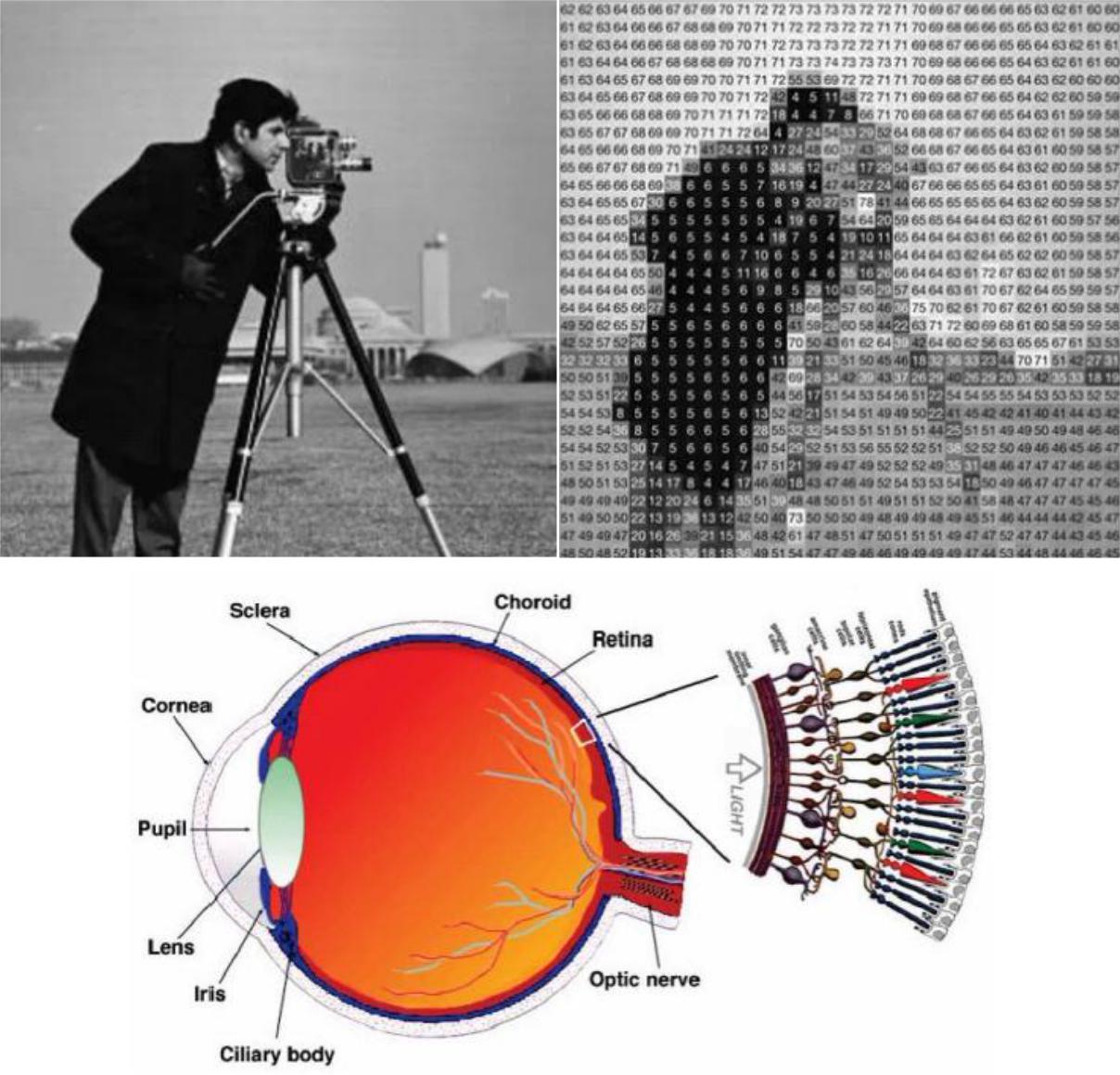
人工智能篇



**3.1 计算机视觉**

3.1.1 什么是计算机视觉

解读 w\*h\*3 个 0～255 之间的数字中蕴含的、人类可理解的内容。

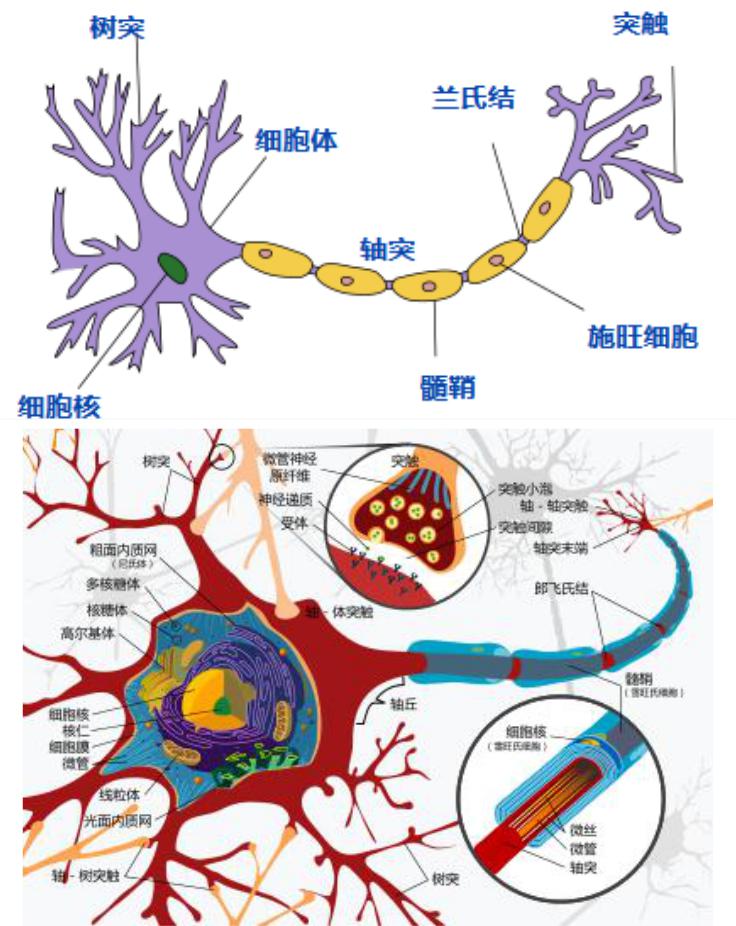


3.1.2 人类视觉的启示

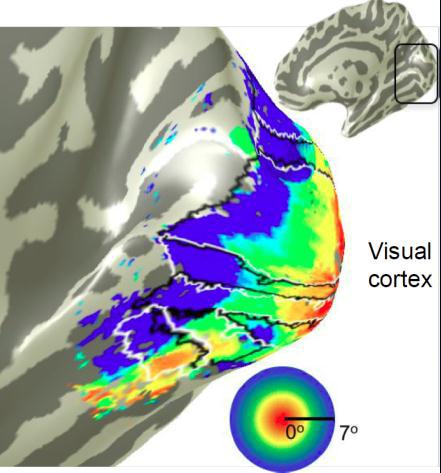
感受野：这个位置里适当的刺激能够引起该神经元反应的区域。

68

人工智能篇



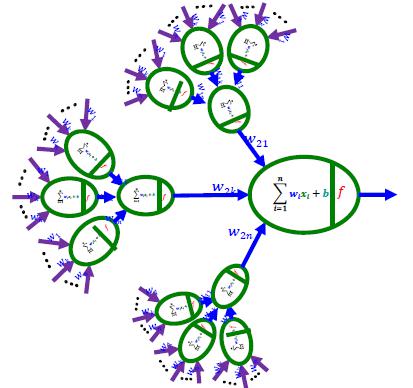
层级感受野：一个神经细胞看的更远（视野更大）、能处理更负责的任务。



神经网络从输入到输出，中间有多个隐藏的层。

69

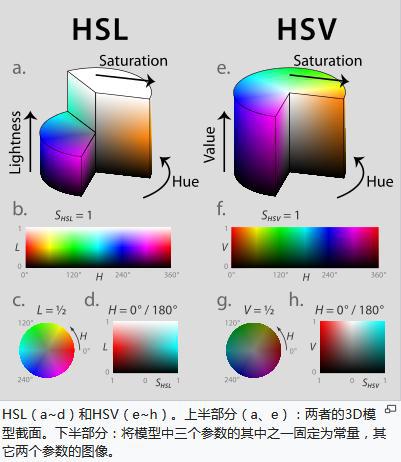
人工智能篇



3.1.3 HSV 和灰度图

HSV 是一种将 RGB 色彩模型中的点在圆柱坐标系中的表示法。这两种表示法试图做到比 RGB 基于笛卡尔坐标系的几何结构更加直观。

* 色相（H）是色彩的基本属性，就是平常所说的颜色名称，如红色、黄色等。
* 饱和度（S）是指色彩的纯度，越高色彩越纯，低则逐渐变灰，取 0-100%的数值。
* 明度（V），亮度（L），取 0-100%。



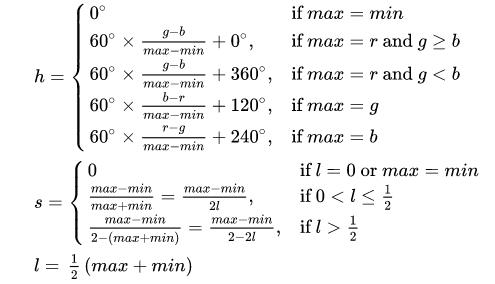
HSV 模型通常用于计算机图形应用中。在用户必须选择一个颜色应用于特定图形元素各种

70

人工智能篇



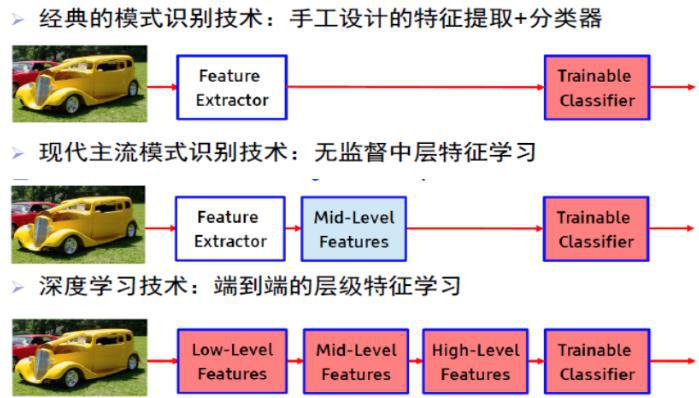
应用环境中，经常使用 HSV 色轮。在其中，色相表示为圆环；可以使用一个独立的三角形来表示饱和度和明度。典型的，这个三角形的垂直轴指示饱和度，而水平轴表示明度。在这种方式下，选择颜色可以首先在圆环中选择色相，在从三角形中选择想要的饱和度和明度。



图像的灰度化：在计算机领域中，灰度（Grayscale）数字图像是每个像素只有一个采样颜色的图像。这类图像通常显示为从最暗黑色到最亮的白色的灰度，尽管理论上这个采样可以是任何颜色的不同深浅，甚至可以是不同亮度上的不同颜色。

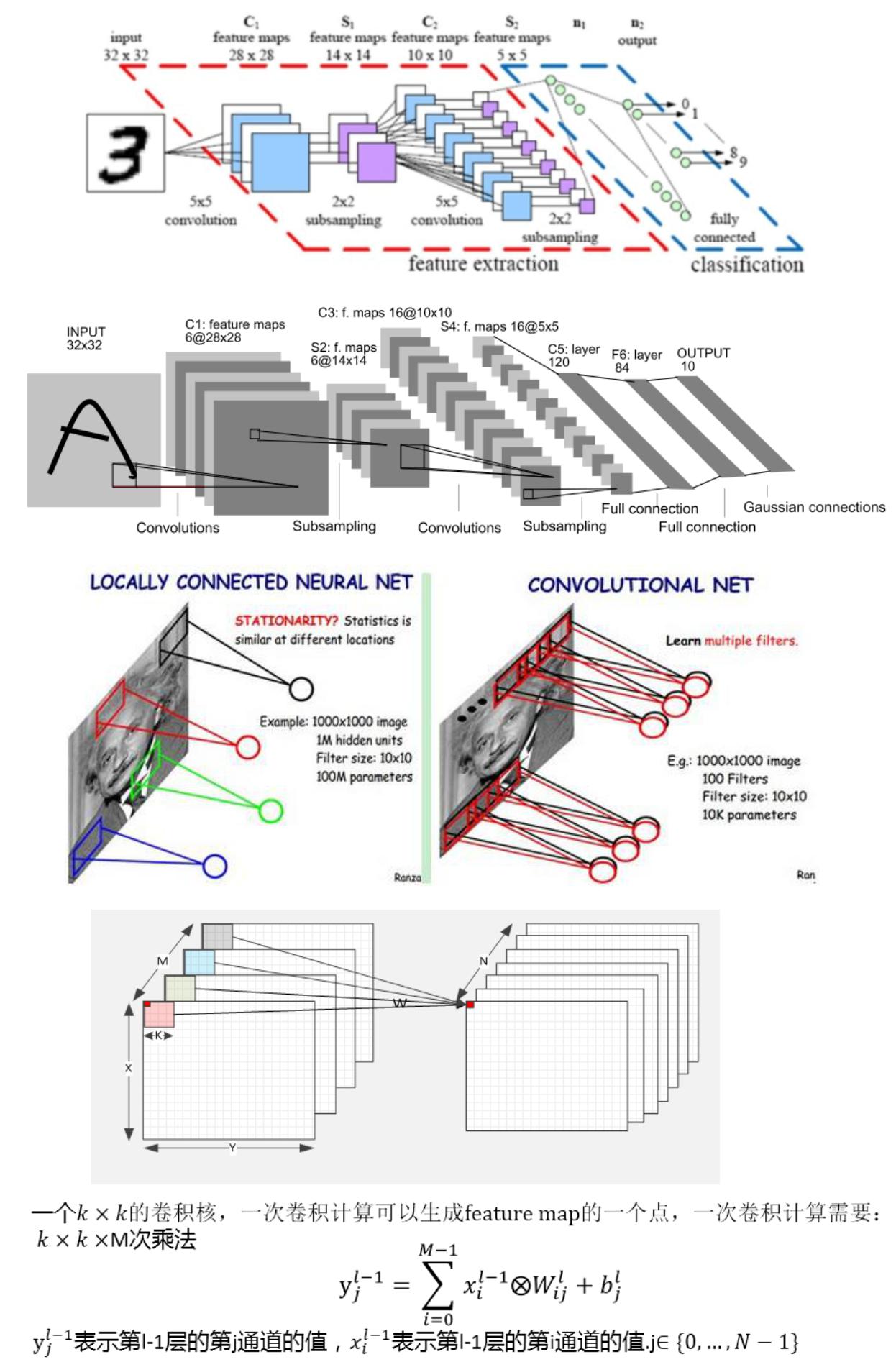
灰度图像与黑白图像不同，在计算机图像领域中黑白图像只有黑白两种颜色，灰度图像在黑色与白色之间还有许多级的颜色深度。但是，在数字图像领域之外，“黑白图像”也表示“灰度图像”，例如灰度的照片通常叫做“黑白照片”。在一些关于数字图像的文章中单色图像等同于灰度图像，在另外一些文章中又等同于黑白图像。

**3.2 基于卷积神经网络的深度学习模型**

****

71

人工智能篇



**3.3 二值化和池化**

72

人工智能篇



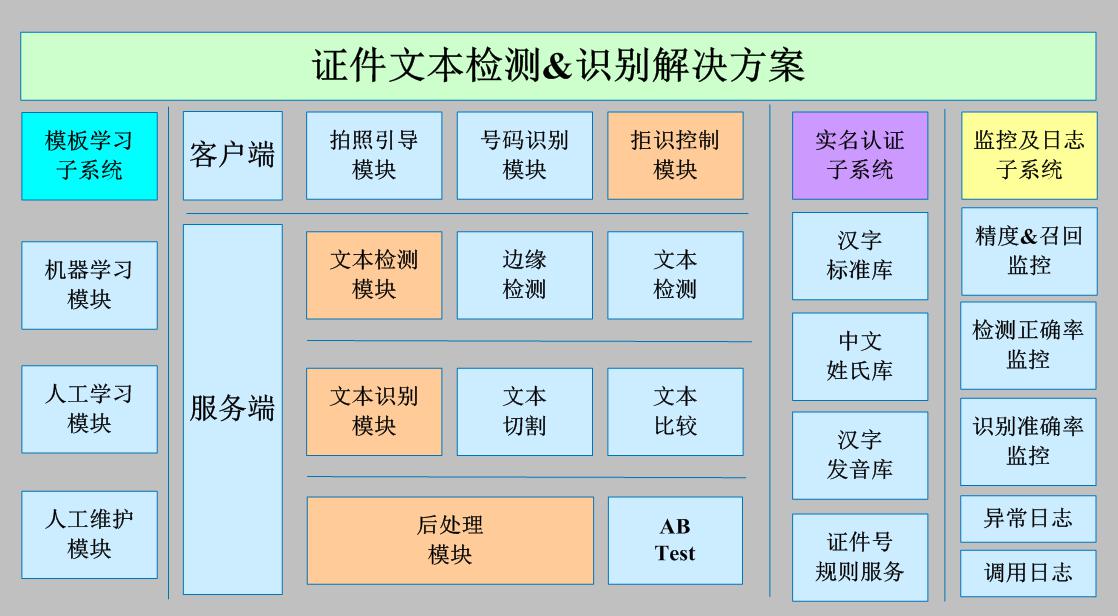
二值化（英语：Thresholding）是图像分割的一种最简单的方法。二值化可以把灰度图像转换成二值图像。把大于某个临界灰度值的像素灰度设为灰度极大值，把小于这个值的像素灰度设为灰度极小值，从而实现二值化。

根据阈值选取的不同，二值化的算法分为固定阈值和自适应阈值。比较常用的二值化方法则有：双峰法、P 参数法、迭代法和 OTSU 法等。



**四、证件 OCR 的架构及实现**

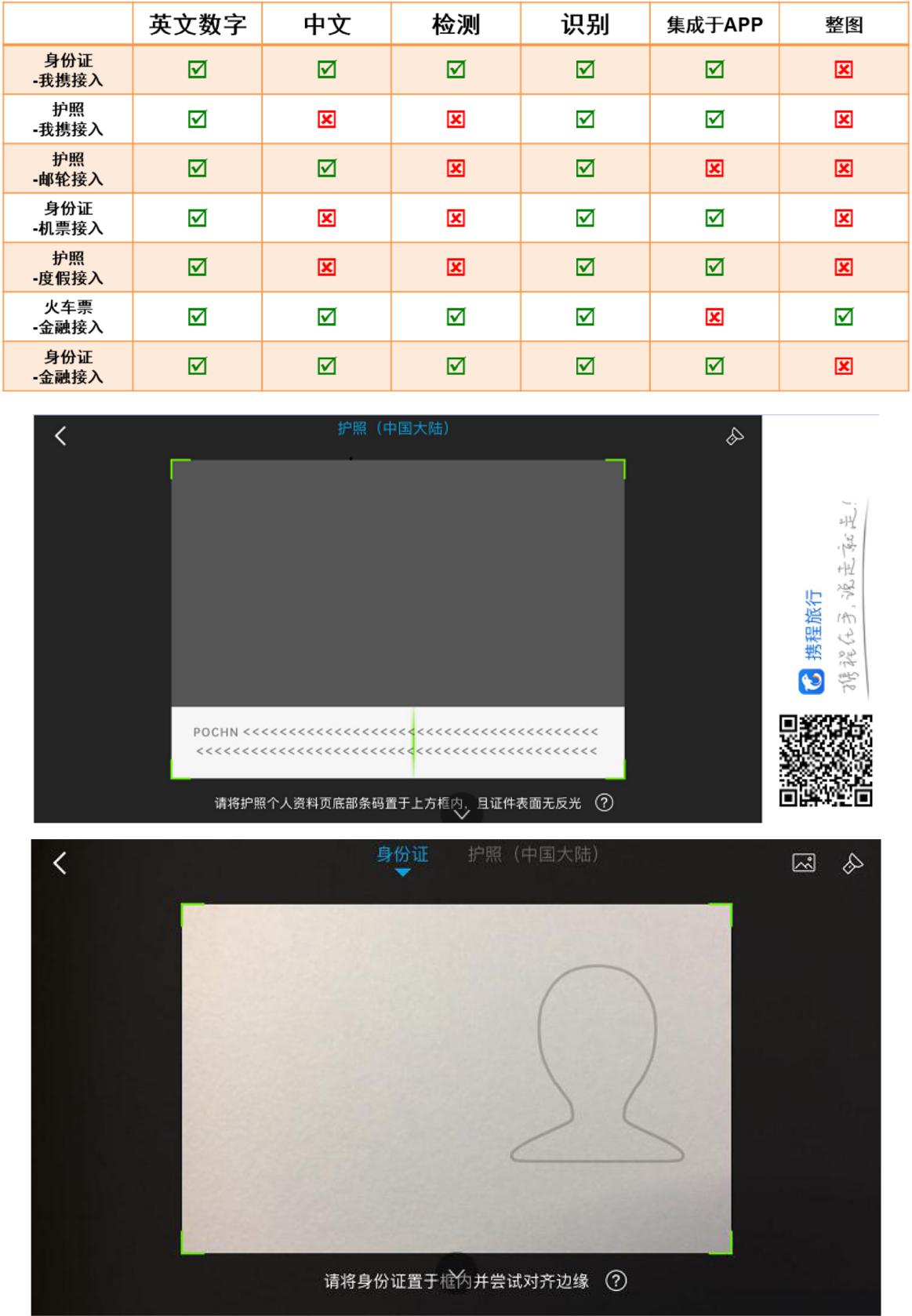
**4.1 架构图**

****

**4.2 应用场景**

73

人工智能篇



**4.3 拒识检测**

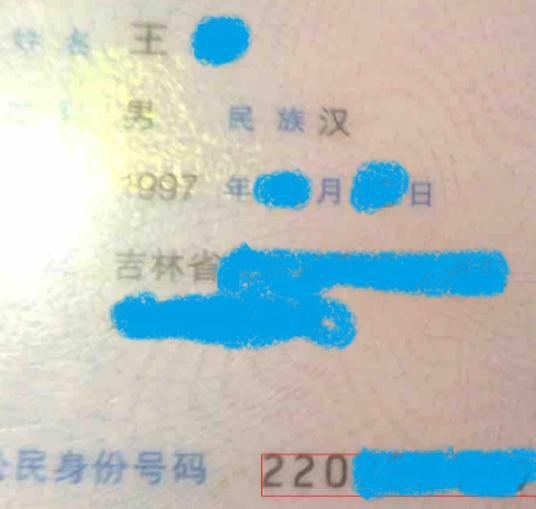
以下图为例，当用户将我们待识别区域（即姓名位置）遮挡时，我们会执行拒识处理。

74

人工智能篇



同理如下图，待识别区域发生明显曝光时，我们也会加入拒识处理。



拒识处理使用直方图均衡等技术。

直方图均衡化是图像处理领域中利用图像直方图对对比度进行调整的方法。

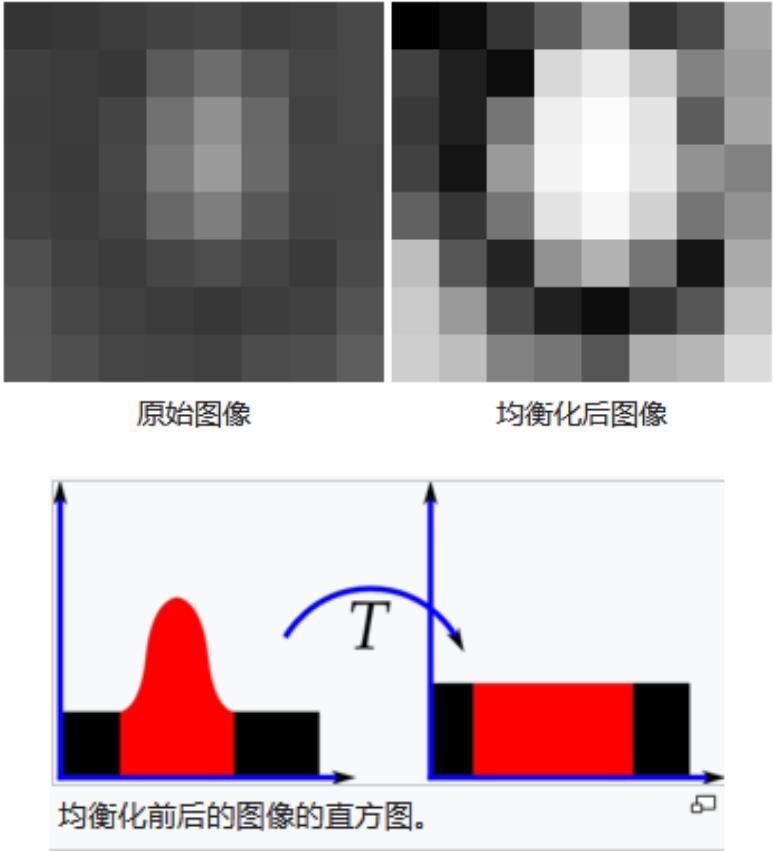
这种方法通常用来增加许多图像的全局对比度，尤其是当图像的有用数据的对比度相当接近的时候。通过这种方法，亮度可以更好地在直方图上分布。这样就可以用于增强局部的对比度而不影响整体的对比度，直方图均衡化通过有效地扩展常用的亮度来实现这种功能。

这种方法对于背景和前景都太亮或者太暗的图像非常有用，尤其是可以带来 X 光图像中更好的骨骼结构显示以及曝光过度或者曝光不足照片中更好的细节。

一个主要优势是它是个相当直观的技术并且是可逆操作，如果已知均衡化函数，那么就可以恢复原始的直方图，并且计算量也不大。一个缺点是它对处理的数据不加选择，它可能会增加背景噪声的对比度并且降低有用信号的对比度。

75

人工智能篇



拒识处理在客户端或页面前端完成，没有前端的场景则在后端完成。拒识处理还使用二分搜索算法通过大量样本判断目标图片是否拒识。

defbinary\_search(arr,start,end,hkey):

ifstart > end:

return-1

mid= start + (end - start) / 2

ifarr[mid] > hkey:

returnbinary\_search(arr, start, mid - 1, hkey)

ifarr[mid] < hkey:

returnbinary\_search(arr, mid + 1, end, hkey)

returnmid

线上版本拒识的精度在 98%-99%之间，拒错率在 20%以下。

**4.4 文本检测**

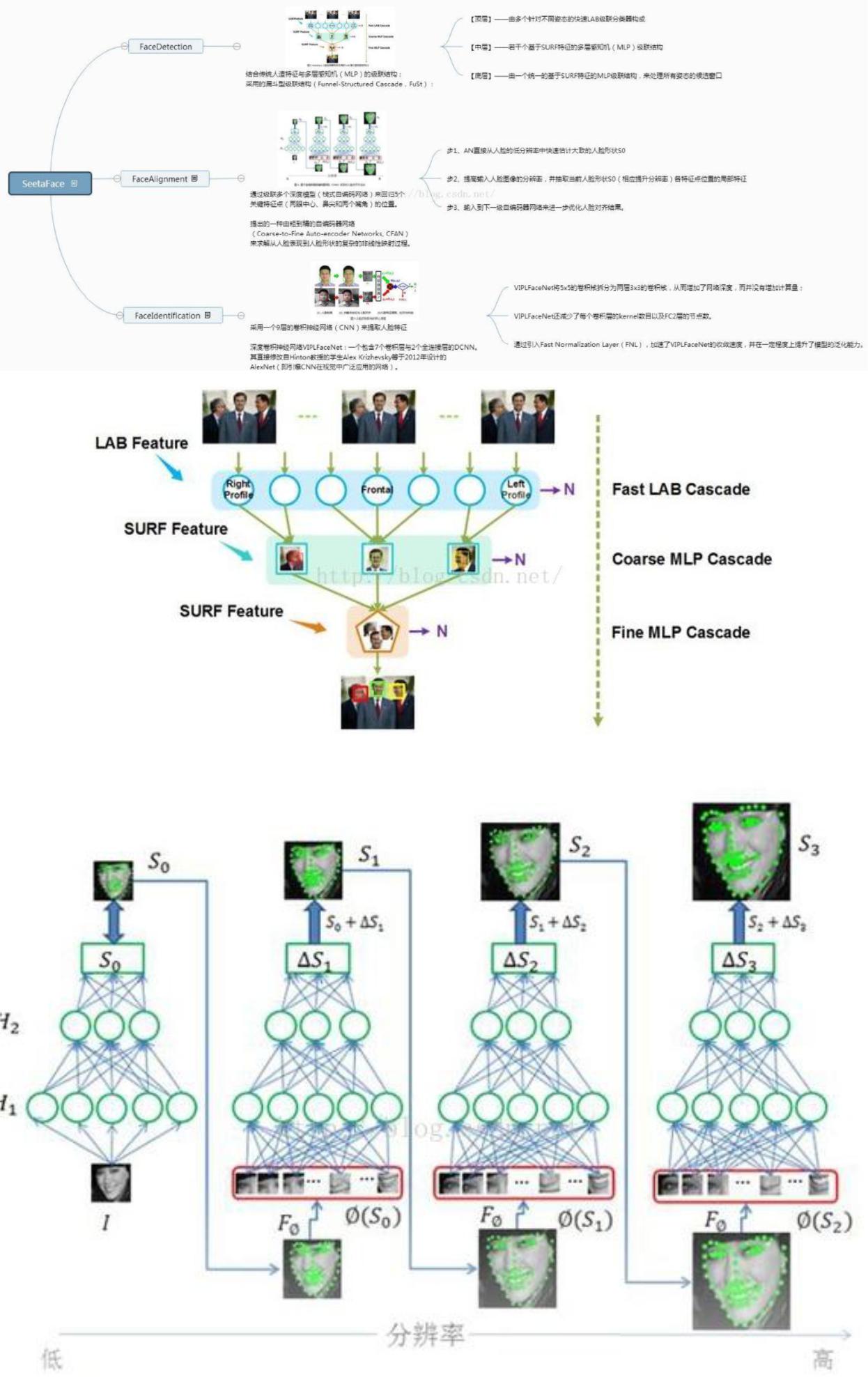
文本检测分为有引导和无引导两类，有引导的文本检测，我们使用先验知识（比如人脸，证件边缘等）和大量样本深度学习目标函数定位待识别区域。

而无引导的情况，则完全使用基于 Attention 的整行识别技术做范文本处理。

人脸识别部分，借鉴了山世光老师的开源项目 seetaface/SeetaFaceEngine，并针对我们证件场景做了一些定制开发。

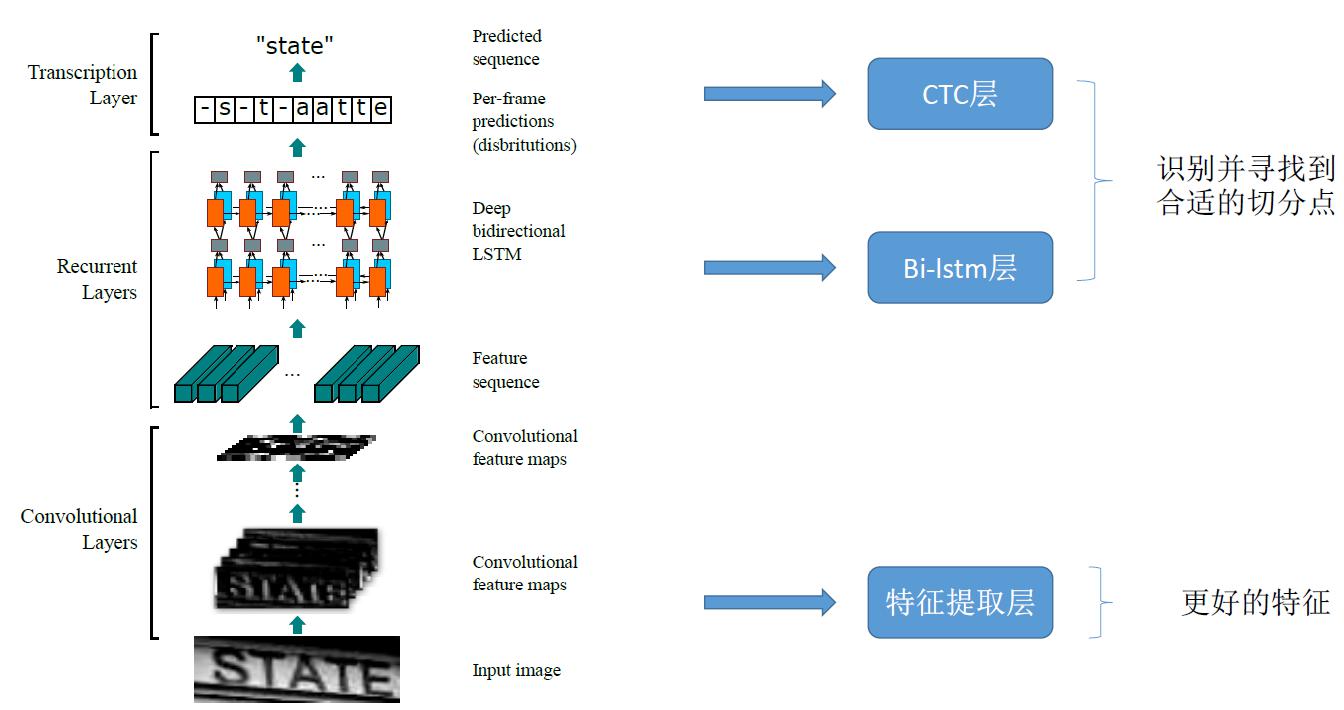
76

人工智能篇



77

人工智能篇



**4.5 文本识别**

文本识别部分我们使用灰度投影在切割无关信息（比如少数民族的拼音、外籍护照的发音注释等）、二值化归一化/下采样池化等技术做比较识别。并且引入了 HOG、LBP、Haar 等特征的权重机制。

int\* v = NULL;//垂直投影

int\* h = NULL;//水平投影

CvScalar s,t;//投影时矩阵的元素

IplImage\* pBinaryImg = NULL;//二值化后图像

IplImage\* pVerticImg = NULL;//垂直投影图像

IplImage\* pHorizImg = NULL;//水平投影图像

int x,y;//图像像素坐标

v=new int[pBinaryImg->width];

h=new int[pBinaryImg->height];

for(i=0;i<pBinaryImg->width;i++)

v[i]=0;

for(i=0;i<pBinaryImg->height;i++)

h[i]=0;

for( x=0;x<pBinaryImg->width;x++)

{

for(y=0;y<pBinaryImg->height;y++)

{

s=cvGet2D(pBinaryImg,y,x); //t=cvGet2D(paint,y,x); if(s.val[0]==0)

v[x]++; //cvSet2D(paint,y,x,t);

}

78

人工智能篇



}

for( y=0;y<pBinaryImg->height;y++)

{

for( x=0;x<pBinaryImg->width;x++)

{

s=cvGet2D(pBinaryImg,y,x); //t=cvGet2D(paint,y,x); if(s.val[0]==0)

h[y]++;

}

}

pVerticImg = cvCreateImage( cvGetSize(pBinaryImg),8,1 );

pHorizImg = cvCreateImage( cvGetSize(pBinaryImg),8, 1);

cvZero(pVerticImg);

cvZero(pHorizImg);

for(x=0;x<pBinaryImg->width;x++)

{

for(y=0;y<v[x];y++)

{

t=cvGet2D(pVerticImg,y,x); //s=cvGet2D(paint,y,x); //t=cvGet2D(paint,y,x); t.val[0]=255;

cvSet2D(pVerticImg,y,x,t);

}

}

for(y=0;y<pBinaryImg->height;y++)

{

for(x=0;x<h[y];x++)

{

t=cvGet2D(pHorizImg,y,x); //s=cvGet2D(paint,y,x); //t=cvGet2D(paint,y,x); t.val[0]=255;

cvSet2D(pHorizImg,y,x,t);

}

}

4.5.1 HOG 特征

方向梯度直方图（Histogram of Oriented Gradient, HOG）特征是一种在计算机视觉和图像处理中用来进行物体检测的特征描述子。它通过计算和统计图像局部区域的梯度方向直方图来构成特征。Hog 特征结合 SVM 分类器已经被广泛应用于图像识别中，尤其在行人检测中获得了极大的成功。需要提醒的是，HOG+SVM 进行行人检测的方法是法国研究人员 Dalal 在 2005 的 CVPR 上提出的，而如今虽然有很多行人检测算法不断提出，但基本都是以 HOG+SVM 的思路为主。

79

人工智能篇



4.5.2 LBP 特征

LBP（Local Binary Pattern，局部二值模式）是一种用来描述图像局部纹理特征的算子；具有旋转不变性和灰度不变性等显著的优点。它是首先由 T. Ojala, M.Pietikäinen, 和 D. Harwood 在 1994 年提出，用于纹理特征提取。而且，提取的特征是图像的局部的纹理特征。

4.5.3 Haar 特征

Haar-like 特征最早是由 Papageorgiou 等应用于人脸表示，Viola 和 Jones 在此基础上，使用 3 种类型 4 种形式的特征。

Haar 特征分为三类：边缘特征、线性特征、中心特征和对角线特征，组合成特征模板。特征模板内有白色和黑色两种矩形，并定义该模板的特征值为白色矩形像素和减去黑色矩形像素和。Haar 特征值反映了图像的灰度变化情况。例如：脸部的一些特征能由矩形特征简单的描述，如：眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深，嘴巴比周围颜色要深等。但矩形特征只对一些简单的图形结构，如边缘、线段较敏感，所以只能描述特定走向（水平、垂直、对角）的结构。

**4.6 一些后处理的思路**

在识别主体逻辑外，我们根据实际场景加入了一些后处理逻辑。

比如身份证号验证、护照号验证、汉字权重验证、中文姓氏库验证、中文发音验证等。

**五、参考资料**

1）https://zh.wikipedia.org/wiki/(2018)

2）Learning2See\_VAI\_Progrss&Trends - 山世光

3）Gradient-Based Learning Applied toDocument Recognition in Proceedings of the IEEE,

1998

4）ImageNetClassification with Deep Convolutional Neural Networks

5）Noteson Convolutional Neural Networks

6）Practical Recommendations for Gradient-BasedTraining of Deep Architectures

7）Stochastic Pooling for Regularization of DeepConvolutional Neural Networks

8）Maxout Networks

【注】文中所列代码并非项目实际代码，都为说明算法需要的开源代码。

80

人工智能篇



**携程 AI 模型引擎设计与实践**

**[作者简介]**李媚，酒店数据智能组应用开发工程师。2016年加入携程，先后负责了酒店交叉推荐，优选频道个性化酒店排序等服务开发工作。

**前言**

近年来人工智能的发展成果在互联网行业得到了广泛的推崇和应用，各大巨头纷纷借助 AI 打造个性化、精细化服务, 加速扩张生态领域。从推荐系统到实时风控，从广告系统到图像处理，模型服务在携程各个业务领域发挥着日益重要的作用。然而回顾现有的模型上线模式，不难发现仍存在一定的缺陷：

1、训练数据准备工作需要手工完成。数据清洗和特征挖掘是模型训练的前期工作，既包括从原始数据清洗出特征数据，也包括对清洗出的特征进行处理。由于缺乏统一的特征管理平台，目前训练需要的原始数据仍需算法工程师自行收集、整理、清洗。

2、不少模型处于离线预测阶段。相对于离线预测，实时预测能够依据用户的实时行为和最新的数据信息作出下一步预测，有效提高预测的准确性。但实时数据存在复杂、多变等特性，以及实时预测对性能上的要求更加严苛，工程技术门槛高，不少团队选择了相对容易实现的离线预测方式。

3、实时模型服务的开发周期长。实时模型服务离不开实时特征准备、业务逻辑开发、模型调用开发等步骤。实时特征一般由各项目的开发工程师自行维护，不可避免地存在特征重复开发的现象，带来开发资源和存储资源的浪费。此外，一个预测场景一般由一个模型服务提供支持，新的模型服务需求需要完全从头开始开发，开发周期较长。

结合上述现状以及在酒店个性化推荐、信息与图片等领域积累的丰富的模型上线经验，携程数据服务组推出了模型引擎平台——旨在通过搭建一个综合性的模型服务平台减少模型上线过程中的重复工作，实现模型的快速上线迭代，健全线上模型的监控评估机制。

**一、平台构建**

**1.1 设计目标**

作为一个综合型的平台，模型引擎致力于提供从数据处理、模型训练到模型上线的全闭环服务，为此我们制定了如下目标：

1）服务于产品经理、数据科学家、开发工程师、测试工程师团队，通过服务全景图的形式串联各环节；

2）作为一个实时预测平台，模型引擎服务接受秒级延迟的实时数据，毫秒级地返回预测结果；

81

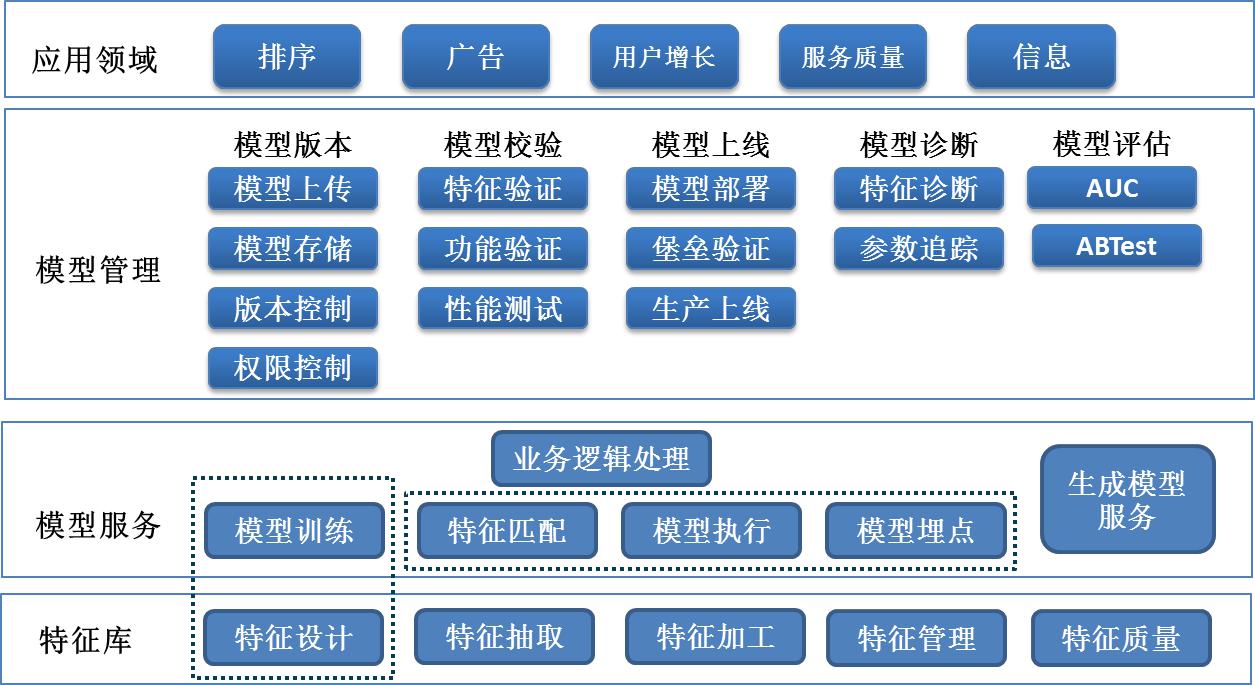
人工智能篇



3）广泛地适用于携程各类业务场景，支持包括 ABTesting、模型灰度上线、多模型融合等功能。

**1.2 总体架构**

针对上述设计目标，模型引擎设置了特征管理、工程管理、模型管理、产品管理四大模块，总体逻辑架构图如下：



1.2.1 特征管理

特征管理模块主要解决特征准备问题，对应了逻辑架构图中的特征库一层。

根据使用场景，特征分为离线特征和在线特征。离线特征主要在模型训练阶段使用，数据科学家对存储在 Hdfs 中的静态数据、落地的消息数据等尝试不同的样本选择、变换处理和组合等，再配合算法及参数进行模型训练。训练完成后，数据科学家将最终的特征方案交给开发工程师，再由开发工程师将离线特征搬移到在线环境，如 Redis，ES，Aerospike 等。

在线特征开发工作既包括将离线存储中的特征导入到实时存储，也包括对实时消息（Kafka、Qmq），例如用户点击、下单、酒店起价变化等进行处理，不断更新在线数据。

模型引擎维护了一个公共的特征库，支持算法工程师通过界面操作的方式添加离线特征，复用已有特征；同时也实现了通过配置自助导出离线特征到实时环境的功能和对流信息的实时处理功能，接入方借助提供的 API 就可以进行数据访问。

同时，对重要特征进行监控与有效性分析是保障模型服务质量的首要环节。相对于离线特征，实时特征有时效性和一致性等要求；而且实时特征一般以 KV 形式存储，不能简单地通过 SQL 语句进行统计。如果每个工程师单独为自己维护的特征开发监控 Job，那么开发工作量就会

82

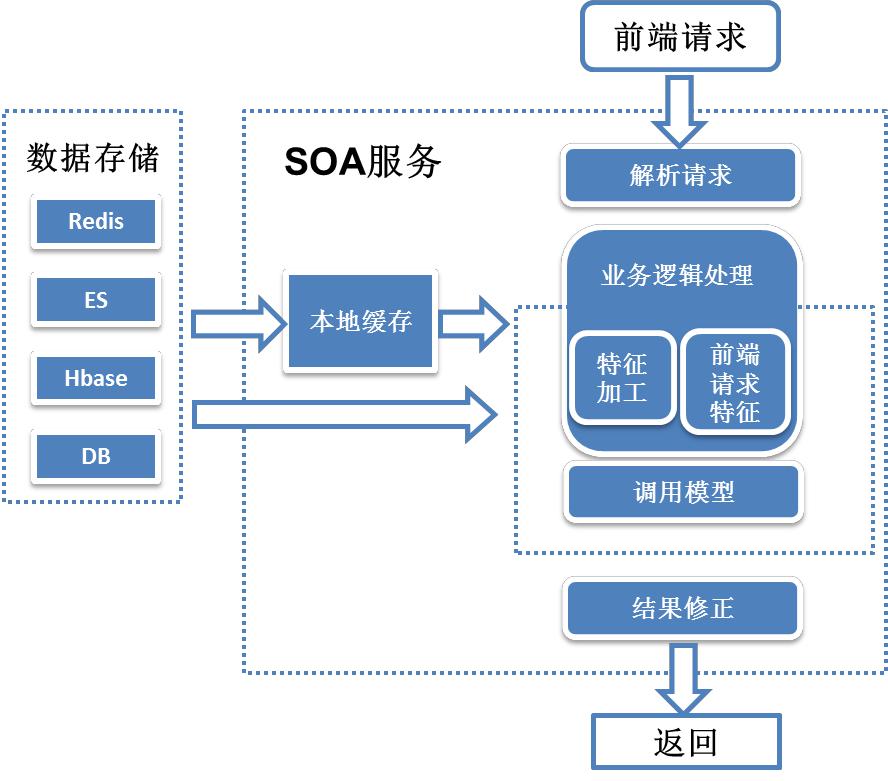
人工智能篇



大大加重。模型引擎由于维护了公共的特征库，可以对特征池中的特征建立了统一的长效监控机制，常见的监控指标，如缺失率、新鲜度、活跃度等，都可以通过模型引擎平台查看和增添。

1.2.2 工程管理

工程管理模块主要解决模型的工程实现问题，对应了逻辑架构图中的模型服务一层。在实时场景中，模型服务一般以 SOA 服务形式供使用方调用，包含了建立本地缓存、实现业务逻辑、调用模型、返回预测结果等步骤，如下图所示。



针对这些典型的开发步骤，模型引擎提供了通用的组件辅助开发工程师，降低工程实现难度。

在实时批量预测场景中一次调用需要用到大量的原始数据，从外部存储获取数据难以保证服务性能，因此工程师会选择建立本地缓存减少数据拉取时间。然而维护本地缓存会带来了不少额外负担，例如不同数据源对新鲜度的要求不同，本地缓存的更新机制要区分设计；本地缓存数据量大时，需要考虑 JVM 参数的配置，以避免频繁的 Full GC 和长时间的 GC。为此，模型引擎开发了 Feature Bus 组件，通过统一的 XML 配置，支持自动构建本地缓存，同时也屏蔽了底层存储的读取细节，免去繁琐的本地缓存开发工作。

在调用模型方面，模型引擎提供了 Code Gen 服务，支持模型服务的自助生成。各模型采用的算法虽然林林总总，但在工程上调用的方式却不外乎几种，例如 PMML、Python 包、 Xgboost 等等。模型引擎对常用的调用方式进行了封装，供开发工程师在项目中直接使用。一方面开发工程师不再需要对没有接触过的模型调用方式进行摸索，另一方面对业务逻辑和模型调用进行了解耦，后续算法升级只需要简单地升级模型服务的版本。而且在生成的模型服务中嵌入了统一的监控，对线上模型服务的质量可以进行集中的管理。

83

人工智能篇



此外，最终进入模型的待预测数据也是算法工程师所关心的。不少原始数据经过了开发工程师手动编码加工，在遇到模型预测效果与线下不一致时首先需要确定这部分数据的正确性。在模型服务中会收集这部分数据并上传，配合平台提供的可视化界面供算法工程师和测试工程师判断数据的健康情况。进一步地，也可以利用这些数据实时更新模型，实现自动化模型训练。

最后，模型拆分、多模型融合功能也在这一层支持。总之，开发工程师只需要将主要精力放在业务逻辑的实现上，以及借助组件串通完整流程。

1.2.3 模型管理

模型管理模块主要实现了模型文件的上传、关联特征、模型发布、迭代回退等功能，对应了逻辑架构图中的模型管理一层。

结合携程的文件服务系统，算法工程师可以在模型管理模块便捷地上传模型。在迭代模型时，为了减少再次测试、发布的环节，模型引擎平台支持算法工程师更替、回退模型文件，实现模型文件的实时生效。在操作的安全性方面，在上传新模型之后，平台会根据样本输入输出数据进行自动校验，确保模型文件的正确性；也会对模型的调用进行压测，符合预设结果的模型才可以进入部署阶段。此外，因为模型文件一旦发布就会即刻生效，为了避免误操作，由测试工程师进行最后的审核上线操作。

现有的机器学习中很多模型的使用都类似黑盒，尽管经过了离线数据的训练以及测试环境的验证，对于模型在实时环境中的预测效果依然难以预料。为此，模型管理模块提供了模型灰度上线功能，支持在正式对外发布前对模型进行内部验证。算法工程师只需要简单配置白名单和上传堡垒模型文件就可以在生产环境直观体验新模型的效果，既不需要开发工程师的介入，也不会影响外网用户。

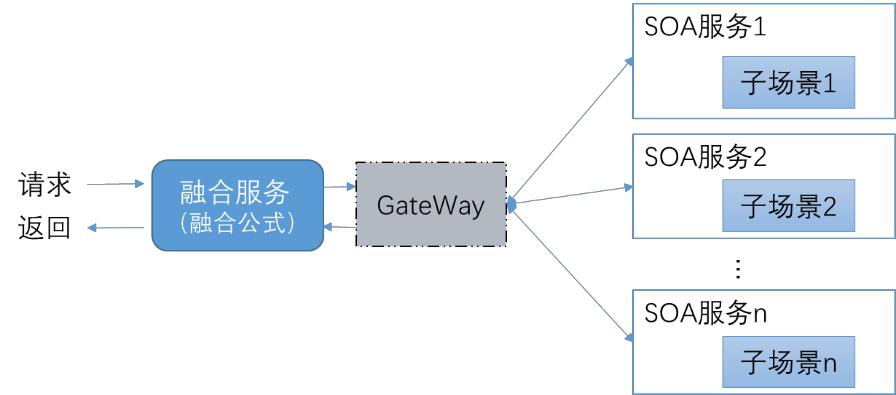
1.2.4 产品管理

产品管理模块主要实现了场景的创建和管理。在模型引擎中一个具体的模型预测业务称为一个场景。项目一般由产品经理发起，所以由产品经理负责创建场景，填写项目信息，并关联相应的开发人员。创建完毕后，算法工程师和开发工程师登录平台就可以在模型管理和工程管理模块下进行操作，为场景实现添砖加瓦。

在模型引擎最初设计中，一个场景仅支持一个模型，根据实验版本不同上传不同的模型文件。然而随着预测业务精细化，在一些场景中，为了达到最优效果，会对模型进行进一步的细分，拆分出多个模型，这些模型的大部分业务逻辑和模型特征都相同，或者模型间存在着依赖关系，因此不适合将这些模型独立为各个场景。因为模型引擎中采用了子场景的概念来支持一个场景下多个模型的情况。其中子场景间的关联关系又分为并行、串行、融合、switch 等多种类型，由此可以支持串行模型，多模型融合等。其中多模型融合的调用方式如下：

84

人工智能篇

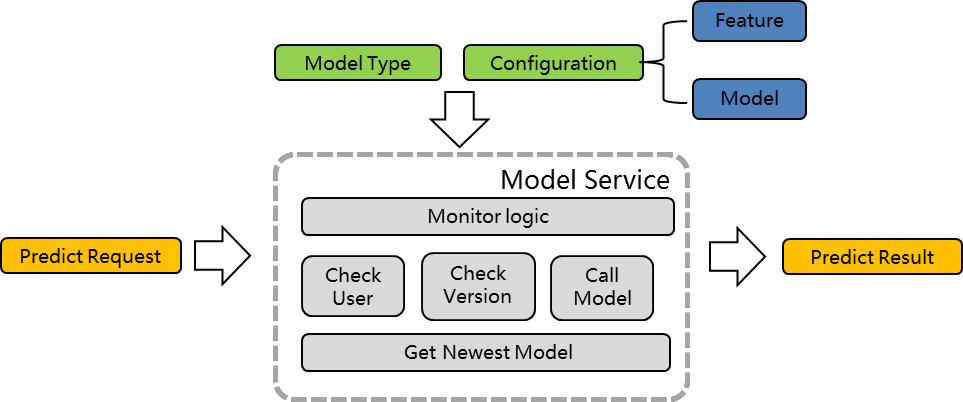


此外，产品管理模块还可以展示模型服务场景的全景图。为了帮助用户清晰地了解各个环节，产品管理关联了一个场景下的特征模块，模型模块，工程实现模块和监控模块，完整串联起了模型上线的整个生命周期，通过点击就可以跳转查看各个模块的明细信息和运行状态；尤其是多子场景的结构展示，可以方便地让相关人员了解整体结构和模型调用链路。

**二、主要组件**

**2.1 Model Service**

Model Service 即 CodeGen 自助生成的模型服务，它支持主流机器学习算法的调用，如 Pmml， Python 包，XGB 等，结构图如下。



当模型确定时，在工程中实现的方式也就确定了，此时开发工程师只需要根据模型的类型，生成对应的模型服务。出于性能上的考虑，目前的实现方式是将生成的服务包引用到工程中，在本地调用。在调用时，将处理好的待预测数据输入，就可以得到预测结果，支持单条或批量调用。当然，模型算法在不断演进中，模型引擎开发组也会负责升级模型服务，此时使用方只需要简单地升级版本即可。

同时，模型服务也封装了最新模型版本的获取功能，自动感知和下载最新的模型文件，实现模型文件更替的分钟级别生效，屏蔽特征不变场景下版本迭代的细节。此外，不同实验版本调用不同模型文件，内测人员和外网用户调用不同模型的支持也由模型服务控制，调用方只

85

人工智能篇



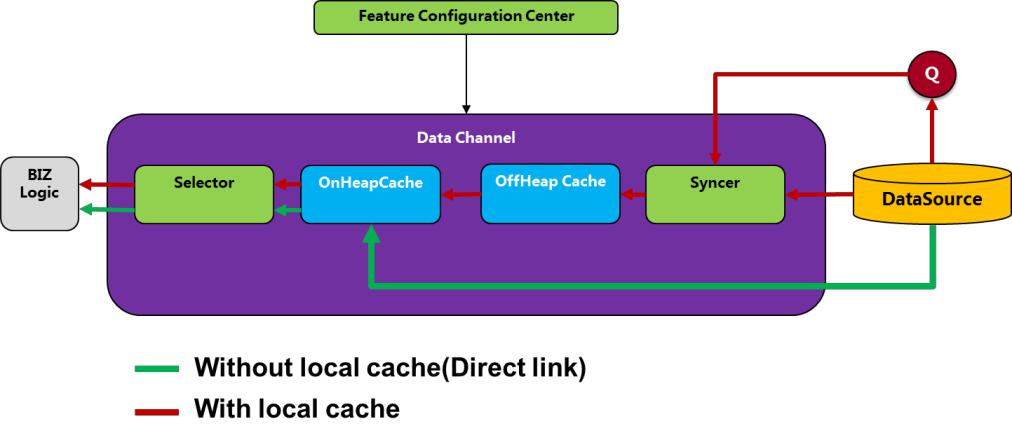
需将实验版本和用户 ID 传入即可。

对待预测数据、预测结果的记录上传和模型服务的监控也由模型服务统一负责。通过规则设定，对上传的数据进行实时判断，及时发现异常。当然开发可以根据各场景的调用量、调用耗时要求选择抽样记录部分数据，避免对存储造成压力。

除了传统意义上的模型，规则和公式也可以认为是广义上的模型。相对于狭义上的模型，公式和规则训练复杂度低，实现简单，在现阶段上线模型代价比较高昂的情况下，是不错的选择。为了支持公式、规则变更等能够实时生效，模型服务也会在后续的版本中增加对其的支持。

**2.2 Feature Bus**

Feature Bus 组件可以协助开发人员自动构建本地缓存，托管数据加载问题，其结构图如下所示。



通过配置 XML 文件指定数据源的存储方式、存储格式、本地缓存格式，结合 Feature Bus 的 jar 包，开发工程师可以轻松地在服务中构建本地缓存。大批量数据缓存在本地容易产生FullGC，进而引起服务响应的暂停，因此 Feature Bus 采取了堆内缓存+堆外缓存的设计方式。堆外内存把内存对象分配在 Java 虚拟机的堆以外的内存，这些内存直接受操作系统管理（而不是虚拟机），能够在一定程度上减少垃圾回收对应用程序造成的影响。对于不需要缓存的轻量级数据，也可以通过 Feature Bus 直接读取。通过统一的 API 方式，开发工程师不再需要为不同的存储方式编写不同的读取代码。

另外，在实时服务中，缓存数据需要保证数据的实时性和一致性。缓存数据的更新又分为批量更新和增量更新。批量更新一般适用于相对静态的数据，例如对酒店静态信息，需要一天更新一次。Feature Bus 支持在 XML 中指定更新的频率，根据实际需求进行批量更新。增量更新数据一般以消息的形式存在，例如用户点击行为，酒店起价变动数据等。在模型引擎的设计中，对这部分数据处理的延迟要求是秒级。Feature Bus 支持在 XML 中配置消息来源、Topic、格式，进而对指定的实时消息进行监听，实时更新本地缓存，有效地将延迟控制在 1 秒以内。不管是批量更新还是增量更新对使用方都是透明的，使用方只需要评估使用场景，进行合理的配置即可。

86

人工智能篇



**三、总结展望**

模型引擎是携程数据服务组对日常开发工作经验的总结和升华，从最贴近实际的场景出发，为模型上线的各环节提供便利。同时，作为一个综合性平台，模型引擎也从特征质量监控、模型调用监控等方面完善了对模型服务质量的把控。

项目自 2017 年年底启动，已经打通了模型上传、审核上线、更新迭代流程，支持 AB 实验，模型灰度验证上线、多模型融合、模型特征在线诊断等功能，接入了酒店排序类、图像识别类、广告类项目等十个项目。

目前模引擎平台已经进入了二期迭代，将在丰富特征库、支持更多类型的模型、完善模型质量在线评估等方面继续发力，并积极推广到其他 BU。

87

人工智能篇



**“猜你所想，答你所问”，携程智能客服算法实践**

**[作者简介]**元凌峰，携程平台中心AI研发部资深算法工程师，负责携程智能客服算法研发，对 Chatbot 相关的 NLP 算法和推荐排序等算法感兴趣。2015 年硕士毕业于上海交通大学图像模式研究所，后加入携程负责实时用户意图和小诗机等项目。

**概述**

作为国内 OTA 的领头羊，携程每天都在服务着成千上万的旅行者。为了保障旅行者的出行，庞大的携程客服在其中扮演着十分重要的角色。但在客服的日常工作中，有很大一部分的行为是重复劳动，这对于客服来说是一种资源浪费。如何从客服工作中解放生产力，提高生产效率成为技术需要解决的一大难题。

随着近几年深度学习算法突破和硬件的升级，人工智能技术在多个领域遍地开花，其中一大应用场景便是智能客服。2017 年初，携程开始大力推进客服智能化的技术研发，目前在酒店售后客服场景，智能客服已经能够解决 70%的问题，不仅提升了客服效率，在服务响应方面也有很大提升。

那么，机器学习或者深度学习在客服领域到底能做什么？怎么做？本文将围绕这两方面介绍携程在智能客服领域中的一些实践经验。

我们先回答这些算法在客服领域到底能做什么。

回答这个问题要回到客服聊天工具这个产品本身，用户在使用客服聊天工具时，最希望的是能够第一时间在客服界面上看到自己想咨询的问题，然后直接找到答案。如果第一眼没有看到想要的问题，那就希望在和“客服”交互过程中以最少的交互次数获取到需要的答案。

在这个过程中，算法不外乎要做的就是两件事：猜你所想，答你所问。

我们先说猜这件事，类似推荐，在用户还没有做出任何输入时，我们会根据用户的信息、当前上下文信息以及咨询的产品信息来猜测用户进入咨询界面时想问什么问题，从而得到一堆问题的排序展示给用户。如果第一步没有猜到用户想要的问题，用户就会通过输入框来简单描述自己的情况和想要咨询的问题，在用户输入的过程中，我们也会结合用户输入的内容通过算法来实时猜测用户可能咨询的问题，并以 input suggestion 的方式给到用户。若上述都无法让用户找到自己想要的答案，那就是答这件事要解决的。

我们会采用 QA 模型对用户发送的话术内容进行分析和匹配，得到用户可能想问的问题，并反馈给用户。这样就完成了一个对话回合，但其实除了上述提到的几个点以外，还有很多地方需要算法参与，比如问题挖掘、关联问题推荐以及上下文对话等等。

88

人工智能篇



下面我们就重点介绍几种常用的算法如何发挥作用。

**一、Question-AnswerMatch**

标准 Q 匹配模型是机器人进行交互的基础模型，对匹配率的要求较高。传统的做法直接根据关键词检索或 BM25 等算法计算相关性排序，但这些方法缺点是需要维护大量的同义词典库和匹配规则。后来发展出潜在语义分析技术（Latent Semantic Analysis，LSA[1,2]），将词句映射到低维连续空间，可在潜在的语义空间上计算相似度。

接着又有了 PLSA（Probabilistic Latent Semantic Analysis）[3]、LDA（LatentDirichlet Allocation）[4]等概率模型，形成非常火热的浅层主题模型技术方向。这些算法对文本的语义表示形式简洁，较好地弥补了传统词汇匹配方法的不足。不过从效果上来看，这些技术都无法完全替代基于字词的匹配技术。

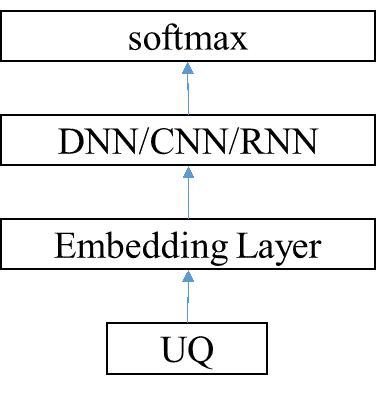
随着深度学习技术兴起后，基于神经网络训练的 Word2vec[5]来进行文本匹配计算引起了广泛的兴趣，而且所得的词语向量表示的语义可计算性进一步加强。但是无监督的 Word2vec 在句子匹配度计算的实用效果上还是存在不足，而且本身没有解决短语、句子的语义表示问题。

因此，研究者们开始研究句子级别上的神经网络语言模型，例如 Microsoft Research 提出的 DSSM 模型（Deep Structured Semantic Model）[6]，华为实验室也提出了一些新的神经网络匹配模型变体[7,8,9]，如基于二维交互匹配的卷积匹配模型。中科院等研究机构也提出了诸如多视角循环神经网络匹配模型（MV-LSTM）[10]、基于矩阵匹配的的层次化匹配模型MatchPyramid[11]等更加精致的神经网络文本匹配模型。虽然模型的结构非常多种，但底层结构单元基本以全链接层、LSTM、卷积层、池化层为主（参考论文[12,13,14,15]的做法）。

**1.1 分类和排序**

在语义模型的训练框架里，大致可以分为两类：分类和排序。

采用分类的方法，一般最后一层接的是多类别的 softmax，即输入是用户 Q，分类结果是所属的标准 Q 类别。



基于分类的模型结构

排序学习有三种类型：point-wise, pair-wise 和 list-wise。在 QA 中我们常用的是 point-wise

89

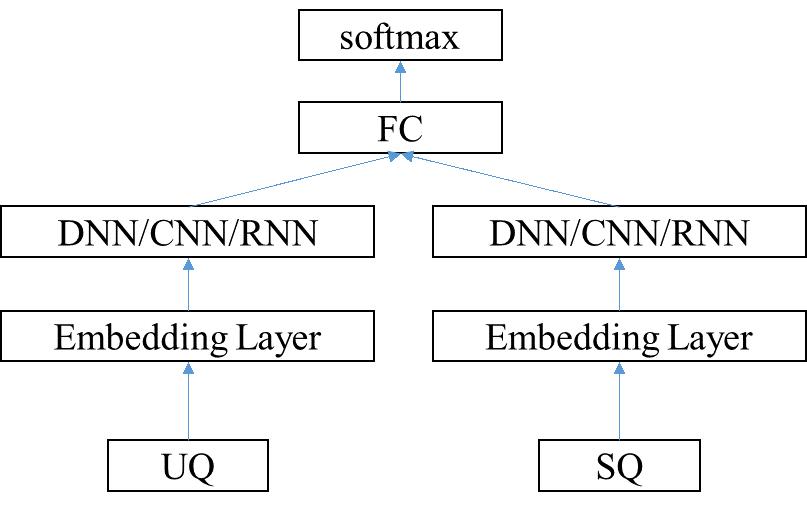
人工智能篇



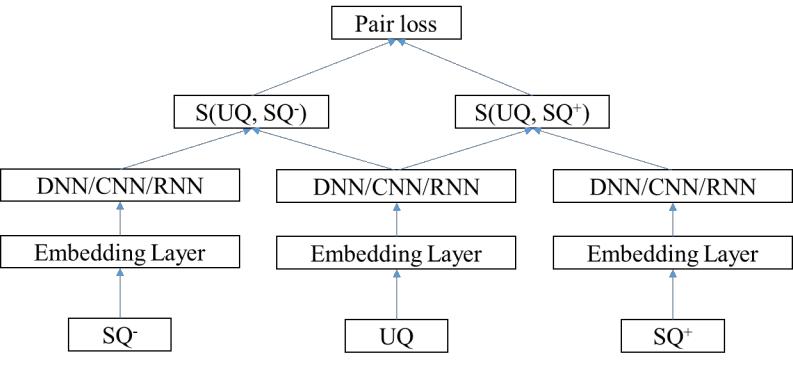
和 pair-wise。其中 point-wise 的方法直接把问题转换成二分类，判断当前用户问题是否属于带匹配的问题，最后根据隶属概率值可以得到问题的排序。而 pair-wise 学习的是和两两之间的排序关系，训练目标是最大化正样本对和负样本对的距离：



其中表示某种距离度量。



基于 point-wise 的模型结构



基于 pair-wise 的模型结构

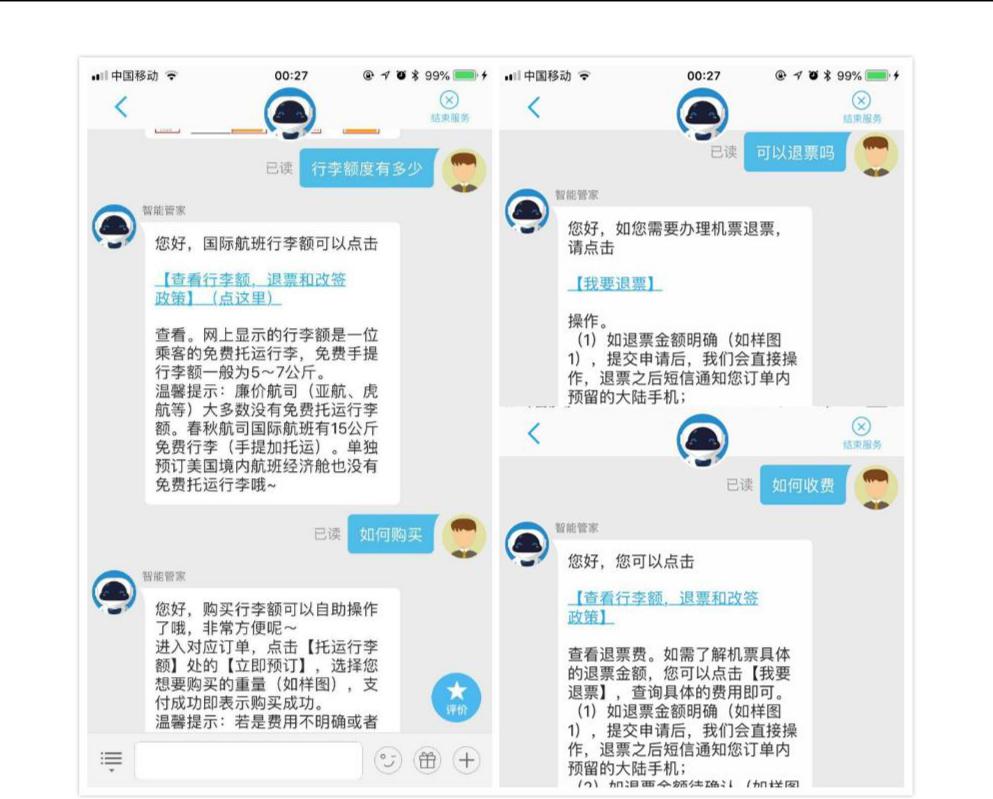
**1.2 单轮和上下文**

一个流畅的会话是需要机器人拥有上下文多轮对话的能力，然而目前大部分 QA 机器人都是基于单轮匹配，如果用户继续上一个问题补充提问，单轮 QA 模型则会断章取义，无法准确识别当前句的准确意图。因此需要把上下文信息进行表示计算，实现多轮匹配模型。

提到多轮对话场景，大家都会想到 Task 任务式对话。但二者在上下文表示方面还是存在一些差异。Task（goal-driven system）是根据预定义的槽位和状态来表示上下文，并且依照某个业务逻辑的对话管理策略（也可以通过 POMDP[16]的方法来构建策略）来引导用户到想要搜索的内容。而 QA（non-goal-driven system）不是面向槽管理的，而是根据用户会话意图来调整对话过程。在 QA 的上下文会话管理方法中，大致可分为两个方向，一个是 Rule-Based[17]的上下文模型；另一个是 Model-Based[18,19]的上下文模型。

90

人工智能篇

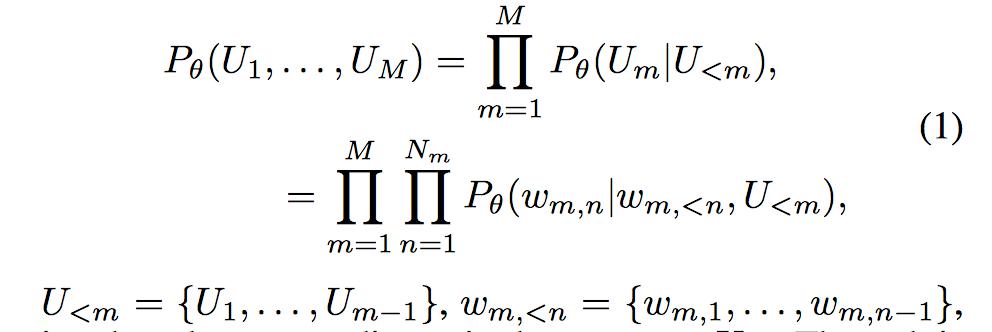


Rule-Based

通过预定义一些先验知识来表示上下文，在会话中不断修改上下文的先验知识并根据上下文记录信息来重排序。

Model-Based

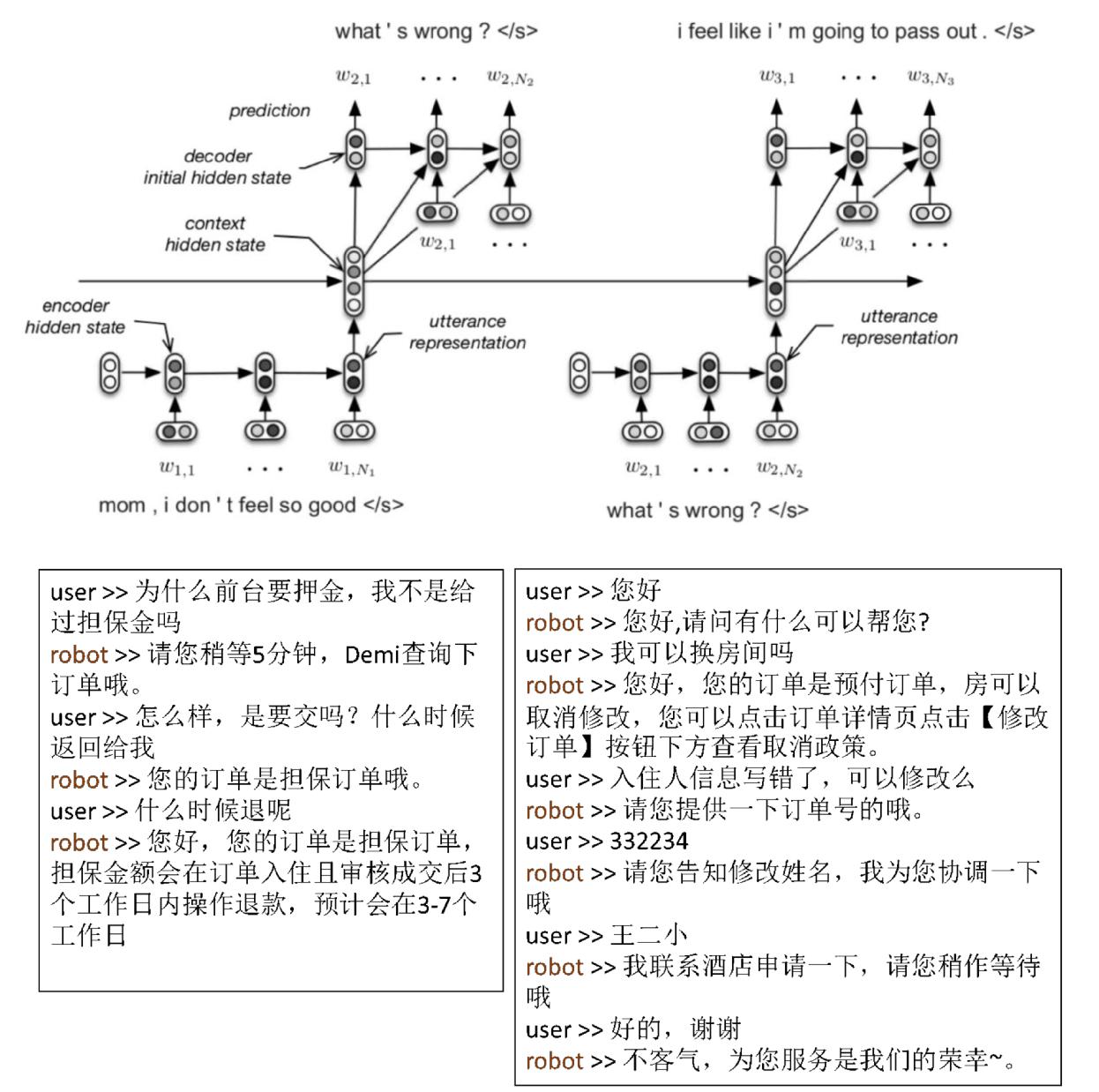
Model-Based 相对于 Rule-Based 的好处就是能够提升泛化能力。有研究者[18,19]就利用模型这种特性，把上下文信息表征在向量里，并通过层次化模型来学习和推断。该模型主要有三个结构：句子级 encoder 模型、context 级别 encoder 模型以及 response decoder 模型。其中 context encoder 模型是解决上下文信息的关键，整个模型用数学模型表示如下：



基于我们的场景对该模型稍作修改并训练后得到效果如下图。

91

人工智能篇



**二、Input suggestion**

输入联想在搜索引擎或者商品搜索中有着十分重要的应用场景，但客服场景的联想提示又区别于搜索，用户的输入是各式各样的，联想结果不一定和用户输入的在词语上完全一致，只需语义上一致即可，而且是能够猜中用户意图。下面聊聊我们在该场景迭代的三种算法。

第一版本采用最常用的 Trie 树结构。字典树的结构简单实用。在搜索中，可以把所有的候选词条建立一个字典树，然后根据用户输入的前缀到 Trie 树中检索候选集，展示给用户。该结构优点是简单有效，能够快速上线。但缺点就是召回率较低，这是因为字典树要求用户输入的词语必须和候选集合里的短语句子要有一致的前缀。对此我们也做了泛化优化，例如去除掉停用词或者无意义词语等，尽可能提高召回。但提升有限。

第二阶段，我们直接采用 point-wise 的排序模型。因为第一阶段的数据积累和人工标注，我们已经有了一定的历史曝光点击数据。考虑到线上联想的请求性能要求较高，我们先尝试了逻辑回归模型。在特征方面，我们主要抽取了三类特征：一类是基于 word2vec 得到的句子特征，另一类是传统的 TF-IDF 特征，最后一类是重要词汇特征（这类特征是通过数据挖掘得到的对应场景的重要词）。该模型上线后整体的使用率比字典树有了明显提升，尤其召回率大幅度提高。

但是，第二阶段的模型仍然存在很多缺陷，我们把线上的曝光未点击数据分析了下，发现如下问题：

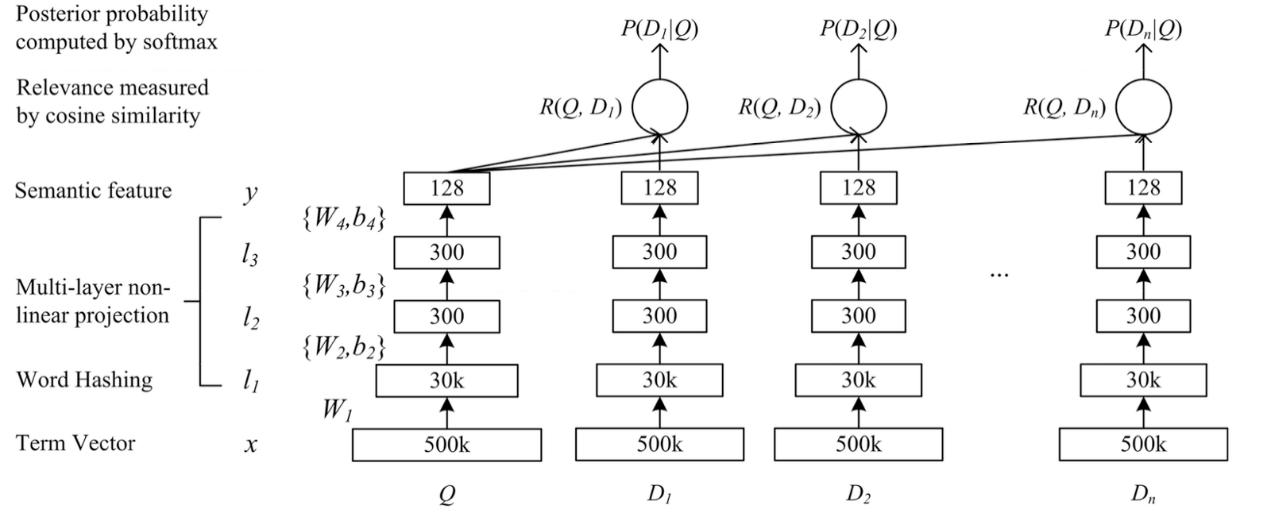
线上存在很多拼音汉字混搭的 case，模型没有解决能力；用户输入的话术存在很多错别字；

联想请求场景以超短文本为主，大都集中在 2-6 个字

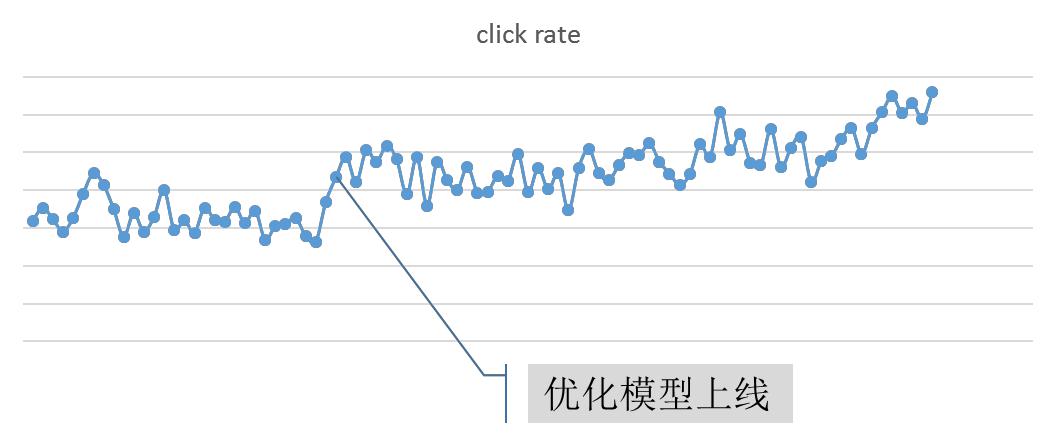
微软的 DSSM 模型在解决短文本语义匹配上有很好的效果，该模型主要亮点引入了 word hashing 操作，该操作能够很好的解决了 OOV（out of vocabulary）问题。其次就是深度神经网络增强了特征的表达能力，该模型计算图如下：

93

人工智能篇



该模型上线后带来显著提升，整个输入联想场景的迭代效果如图。



**三、总结**

限于篇幅，除了上述提到的几种场景，机器学习和深度学习算法还在其他多个场景中辅助携程客服的工作，帮助提升客服工作效率和用户体验。

人工智能并不是新兴的黑科技，只不过近几年深度学习的快速发展让这个词重新活跃在大众视野中。尼尔逊教授对人工智能下了这样一个定义：“人工智能是关于知识的学科――怎样表示知识以及怎样获得知识并使用知识的科学。”

我们更愿意把人工智能看成是人工+智能算法+数据的一个综合体。算法工程师的定义不是简简单单懂算法就可以，而是要懂得如何用算法去优化人工提高效率，如何用算法去挖掘有效信息，紧密地让数据、算法、人工形成一个闭环。

**Reference**

1. DennisS, Landauer T, Kintsch W, et al. Introduction to latent semanticanalysis[C]//Slides from the tutorial given at the 25th Annual Meeting of theCognitive Science Society, Boston. 2003.
2. DeerwesterS, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis[J].Journal

94

人工智能篇



of the American society for information science, 1990, 41(6): 391-407.

1. HofmannT. Unsupervised learning by probabilistic latent semantic analysis[J]. Machinelearning, 2001, 42(1-2): 177-196.
2. BleiD M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machineLearning research, 2003, 3(Jan): 993-1022.
3. MikolovT, Chen K, Corrado G, et al. Efficient estimation of word representations invector space[J]. arXiv preprint arXiv:1301.3781, 2013.
4. HuangP S, He X, Gao J, et al. Learning deep structured semantic models for websearch using clickthrough data[C]//Proceedings of the 22nd ACM internationalconference on Conference on information & knowledge management. ACM, 2013:2333-2338.
5. LuZ, Li H. A deep architecture for matching short texts[C]//Advances in NeuralInformation Processing Systems. 2013: 1367-1375.
6. JiZ, Lu Z, Li H. An information retrieval approach to short text conversation[J].arXiv preprint arXiv:1408.6988, 2014.
7. HuB, Lu Z, Li H, et al. Convolutional neural network architectures for matchingnatural language sentences[C]//Advances in neural information processingsystems. 2014: 2042-2050. [10]Wan,Shengxian, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and XueqiCheng."A Deep Architecture for Semantic Matching with MultiplePositionalSentence Representations." In AAAI, pp. 2835-2841. 2016.

[11]Pang,Liang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and XueqiCheng."Text Matching as Image Recognition." In AAAI, pp. 2793-2799.2016.s

1. FengM, Xiang B, Glass M R, et al. Applying deep learning to answer selection: Astudy and an open task[J]. arXiv preprint arXiv:1508.01585, 2015.
2. LaiS, Xu L, Liu K, et al. Recurrent Convolutional Neural Networks for TextClassification[C]//AAAI. 2015, 333: 2267-2273.
3. SantosC, Tan M, Xiang B, et al. Attentive pooling networks[J]. arXiv preprintarXiv:1602.03609, 2016.
4. KimY. Convolutional neural networks for sentence classification[J]. arXiv preprintarXiv:1408.5882, 2014.
5. YoungS, Gašić M,Thomson B, et al. Pomdp-based statistical spoken dialog systems: A review[J].Proceedings of the IEEE, 2013, 101(5): 1160-1179.
6. LangleyP, Meadows B, Gabaldon A, et al. Abductive understanding of dialogues aboutjoint activities[J]. Interaction Studies, 2014, 15(3): 426-454.
7. SerbanI V, Sordoni A, Bengio Y, et al. Building End-To-End Dialogue Systems UsingGenerative Hierarchical Neural Network Models[C]//AAAI. 2016, 16: 3776-3784.
8. SordoniA, Galley M, Auli M, et al. A neural network approach to context-sensitivegeneration of conversational responses[J]. arXiv preprint arXiv:1506.06714,2015.

95

人工智能篇



**知识图谱在旅游领域有哪些应用？携程度假团队这样回答**

**[作者简介]**鞠建勋，携程度假AI研发团队资深算法工程师，主要负责携程度假自然语言处理相关的 AI 项目。硕士毕业于南京大学，有五年的自然语言处理经验，专注于自然语言处理和知识图谱方面的应用和算法研发。

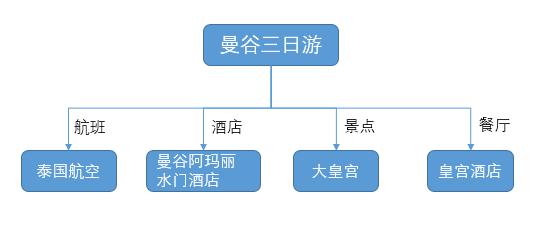
随着互联网和大数据的发展，数据呈现爆炸式增长的态势。知识图谱以其强大的语义处理能力和开放组织能力，为大数据时代的知识化组织和智能应用奠定了基础。

旅游行业作为综合性行业，包含交通、游览、住宿、餐饮、购物、文娱等多个环节，每个环节都有着海量的数据，并且有着相当庞大的应用场景。

同时，旅游领域的数据有着领域范围大，涉及知识面广，知识层级多的特点，比如一个简单的团队游产品就会和飞机、酒店、景点、火车、汽车、餐厅等等多个实体产生关联。对于旅游行业，尤其是互联网旅游行业，如何构建和应用一个旅游领域的知识图谱成为一个非常有价值的问题。

本文将从旅游领域知识图谱的特点，知识图谱的构建，知识图谱的应用三个方面介绍知识图谱在互联网旅游行业的应用。

**一、旅游领域知识图谱的特点**

****

如上图所示，曼谷三日游是一个旅游产品，它包含泰国航空、曼谷阿玛丽水门酒店、大皇宫、皇宫酒店等子产品，每个子产品都有自己不同的数据属性和架构。

泰国航空，包含航班号、出发时间、出发地点、到达时间、到达地点等数据；曼谷阿玛丽水门酒店，包含酒店房型、酒店价格、酒店地址、酒店设施等信息；景点包含门票、地址、景点类型等信息；而餐厅有着菜品、价格、地址等重要信息。

如果这个旅游产品的信息采用传统数据库进行存储，需要多张数据表进行存储，不便于对于复杂数据的整合，也不便于进行一些涉及多领域的推理，而且一旦涉及到复杂的查询，就需

96

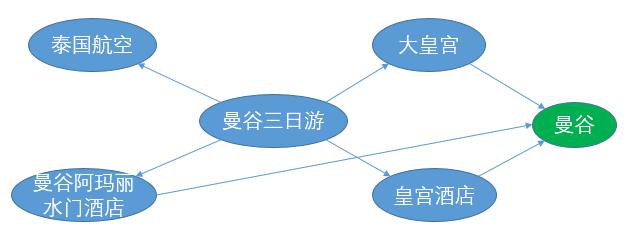
人工智能篇



要多张表的关联，不利于知识的快速获取。

为了解决上述问题，我们构建一个图数据库去存储这些产品的数据。当数据以图的形式存在，就可以发现很多路径存在于两个事物之间，通过路径的跟踪以及一些图的算法就很容易得到一些事物之间的关系。这个图数据库就构成了一个旅游领域的知识图谱。

我们把图数据库中的泰国航空、曼谷阿玛丽水门酒店、大皇宫、皇宫酒店、曼谷、泰国等等设为节点，并称之为实体；把他们之间的关系设为节点和节点之间的边；而把航班号、酒店房型、地址等信息称之为实体的属性。



例如，游客想知道，大皇宫属于哪个国家，通过“大皇宫”->[所属城市]->“曼谷”->[所属国家]->“泰国”的“实体->关系->实体”递推关系就可以很容易找到答案；而传统数据库需要专门为查询国家写一套解决逻辑。又例如，我们想查询一下大皇宫附近的酒店价格，可以通过“大皇宫”->[附近酒店]->“xx 酒店”->[价格]->“xx 元”的关系找到答案；而传统数据库则需要多张表的联合查询才有可能完成这个查询。

此外，知识图谱还有强大的语义概括能力和语义抽象能力。我们可以把实体的类型抽象成Class(类别)，也叫做本体，通过本体与本体之间的关系来完成一些推理或语义分析。

比如我们把大皇宫归于“古代建筑”这个类别，同时定义“古代建筑”属于“建筑”的子类，“建筑”则有“高度”的属性，那么大皇宫也有“高度”的属性。我们不需要给每个子类定义一些通用属性，而只要通过继承父类的属性就可以完成定义。

**二、旅游领域知识图谱的构建**

旅游领域知识图谱的构建，来自于企业数据和外部数据的融合。

首先，构建一个旅游领域的知识图谱，需要企业内部数据作为基础，因为企业内部数据是和企业产品息息相关的，而产品是一切应用的基础。

旅游产品中的信息大部分作为基础数据存在，比如酒店房型，酒店地址等等，我们需要这些信息来构建酒店的知识图谱。而对于一些变化非常频繁的数据，则不会导入到知识图谱中，比如售卖数量等等。我们把这类知识图谱统称为行业知识图谱，它具有高深度和专业性的特点。

此外，我们还需要用到外部数据作为补充，比如我们经常会用到 wiki 上的常识信息来补充知识图谱，我们把这部分知识图谱称为通用知识图谱。

97

人工智能篇

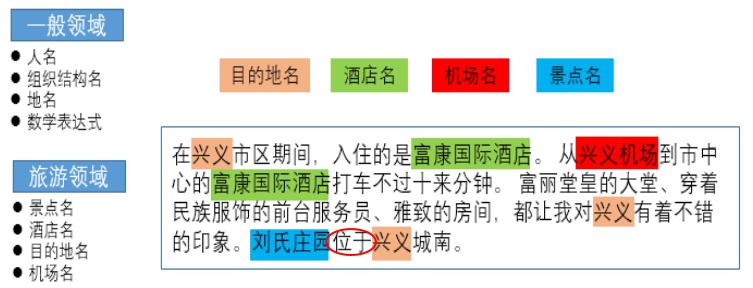


通常常识信息不会存储在企业内部数据库，且具有非常大的广度，比如大皇宫的面积有多大，高度有多高等等。这些常识信息对于产品售卖可能没有太大作用，但对于回答客户的问题或提供搜索依据是有很大帮助的。

通用知识图谱的广度和行业知识图谱的深度相结合，可以互相补充，形成更加完善的知识图谱。

外部数据经常会以大段文本的形式存在，因此我们需要从文本中抽取关系。通常采用自然语言处理的 NER 方法去提取文本中的实体，结合句法分析、远程监督等传统方法来提取实体与实体之间的关系。

如下图所示，我们可以抽取出刘氏庄园这个景点名，同时可以抽取出兴义这个目的地名，也可以挖掘出它们之间的“位于”关系。那么这段文本中的“刘氏庄园”->[位于]->“兴义”这条知识图谱数据就成功被抽取出来了。

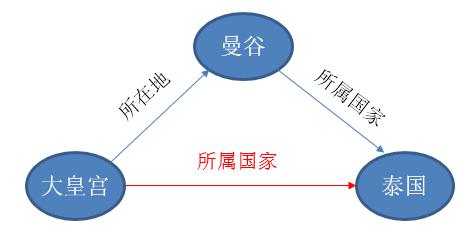


知识图谱的存储可以选择 RDF 数据库，或者图数据库。RDF 数据库是 W3C 标准的以三元组的形式存储的知识图谱，有标准的推理引擎，易于进行推理和关系发现；而图数据库则以节点和边的方式存储知识图谱，节点和边都可以带属性，没有标准的推理引擎，但图数据库的遍历效率高，性能更好，更适合企业级的海量数据。

当构建好旅游领域的知识图谱之后，可以对知识图谱进行补全。比如大皇宫的所属城市是曼谷，而曼谷是泰国的一个城市，假如缺失了“大皇宫”->[所属国家]->“泰国”这条关系，我们可以通过关系补全来把这条关系加入知识图谱。关系补全的方法有很多，如 TransE、 TransH 和 TransR 等，本文就不一一阐述了。

98

人工智能篇



**三、旅游领域知识图谱的应用**

知识图谱的应用很广，结合知识图谱和旅游领域的特点，旅游知识图谱可以应用在以下几个方面：

**3.1 旅游问答机器人**

通常问答机器人不擅长去“寻找”答案，而是需要人来提前“设定”好答案。但有了知识图谱之后，这种情况会得到一些改变。

基于知识库的问答系统我们称之为 KBQA（Knowledge based Question Answering），而知识图谱是知识库的一种。KBQA 更擅长回答“what”“when”“where”问题，只要捕捉到问句中的实体和关系，就可以在知识图谱中“寻找”到相应的实体以及它的属性，而通过实体的属性可以解答大部分的“what”“when”“where”问题。

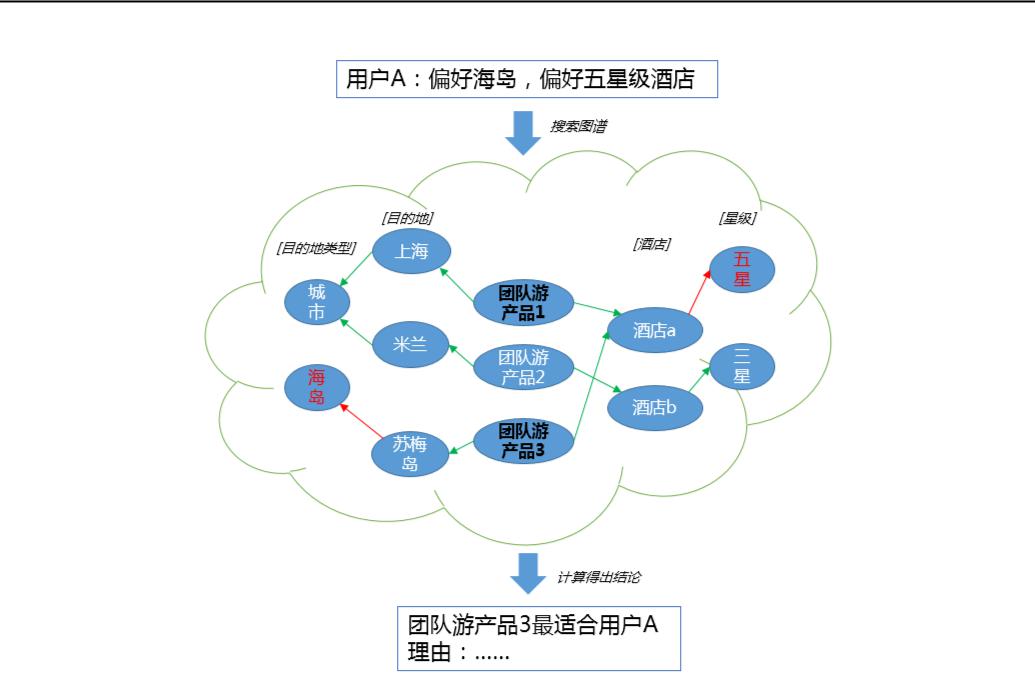
目前有很多问答模型利用知识图谱来解答问题。通常把实体和关系作为一种特征，参与问句意图的计算。我们首先解析问题的语义特征，提取问题中的实体和关系，然后将实体与关系映射到知识图谱中，把知识图谱中对应的实体属性返回作为答案，或者将知识图谱中的实体、关系、属性作为特征，参与下一步的模型计算。

**3.2 旅游内容推荐**

旅游产品非常多样化，所以很依赖推荐。我们往往会抓住旅游产品和用户的浅层特征去推荐，而忽略了一些深层次的特征。当用户和某个知识图谱实体产生关联，我们可以用知识图谱去补充用户特征。

99

人工智能篇



如上图所示，当我们通过数据挖掘发现用户 A 的兴趣点是海岛和五星级酒店后，我们可以通过知识图谱关联出相关的团队游产品。知识图谱提供的信息可以作为推荐系统的一个重要维度，参与下一步的计算，为精准推荐增加一块砝码。

**3.3 旅游内容搜索**

旅游产品的数量很多，因此对于搜索的要求很高。对于一些复杂的搜索语句，知识图谱可以利用自身的推理能力搜索出深层的答案。

例如，用户在搜索框输入“大皇宫附近的酒店”，知识图谱可以通过“大皇宫”->[附近酒店]-> “xx 酒店”搜索出相关的酒店。如果用户在搜索框输入“大皇宫附近酒店的价格”，知识图谱可以通过“大皇宫”->[附近酒店]->“xx 酒店“->[价格]->”xx 元“搜索出酒店的价格属性。知识图谱利用了图数据库的优势，非常方便地进行图遍历，加强了搜索引擎解决复杂搜索的能力。

**四、总结**

知识图谱在旅游场景还有很多其他应用，本文就不一一列举了。

知识图谱这项技术还处于发展阶段，还有很大的提升空间，结合日益增长的旅游市场，我们非常看好知识图谱应用于旅游领域的前景。我们相信，未来的知识图谱还会有更多的应用和更多的技术，为互联网旅游行业的发展添砖加瓦。

100

人工智能篇



**如何选出最“美”图片展示给你？携程做了基于深度学习的图像美感评分系统**

**[作者简介]**路婵，携程度假AI研发团队算法工程师，专注于计算机视觉和机器学习的研究与应用。现阶段致力于度假图像智能化，多次参加国内外数据竞赛并获奖。

**概述**

作为 OTA 行业的领跑者，携程每天服务成千上万的客户。景点的图片介绍成为用户了解景区、玩乐产品的重要参考。

为了给用户带来更好的使用体验，携程门票列表页通常由人工筛选更具代表性的优质图片，指定为每个产品的首图。这种人工指定的方式主观性强，费力度大，无法做到准确而及时的更新。面对大量景点图像，如何智能选择更优质的图像，提高用户满意度，改善用户体验，大幅减少图像的人工干预，成为急需解决的问题。

近几年深度学习在图像领域取得了突破性的进展，基于深度学习的应用也层出不穷。深度神经网络对图像在特定目标域拥有极强的感知与决策能力，将其应用到图像美感评价上，可以综合图像的美学与语义信息，自动选择更优质的首图，为用户带来更好的体验。优化前后对比如下：



**一、技术简介**

101

人工智能篇



构建智能美感评分系统，旨在自动筛选更优质的图像进行展示。本文还通过 CAM 可视化方法，对网络的决策进行了解释。在业务处理中，通过识别图像类别，筛选更适合的图片作为首图进行展示。

**1.1 图像美感评分**

图像美感的量化是图像处理和计算机视觉中的一个问题，其主要目的是预测与人类感知相关的质量分数。与图像质量评价（Image Quality Assessment,IQA ）处理的像素级的退化（degradation）问题不同，美感评价提取图像中与情感和美感相关的语义层次特征。

图像美感评价主要分为两个部分：特征提取与决策。

传统特征提取方法通过人工设计的低层特征（颜色、纹理、清晰度等）和高层特征（景深、区域对比度等），作为图像的美感特征，通过训练一个分类器或回归模型，得到图像的美学质量评分。

相较于传统方法，深度卷积神经网络拥有强大的自动特征学习能力，在图像美感评价方面展现出良好的性能，成为解决该问题的主流方法。

1.1.1 研究现状

美感的评估难点主要集中在全局语义与局部信息的特征抽取上。

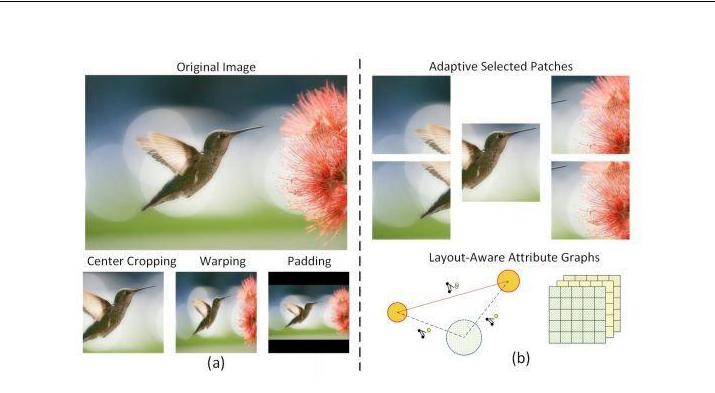
为了兼顾全局语义特征与局部信息，研究者往往采用整张图像与多次裁剪得到的图像片段（multipatch）同时作为输入。

Lu 等的 RAPID 模型[ 1 ]基于 AlexNet，采用多路网络，分别将整张图像与多次随机裁剪得到的部分图像作为输入，提取全局美学特征、局部美学特征，将质量评估作为二元分类问题，对图像美感进行预测。

此外，作者还通过使用图像风格监督（style-column）或者图像语义监督（semantic-column） CNN 合并图像风格信息来进一步提高网络的表现。A-Lamp 架构[ 2 ]采用特定的多次裁剪得到而不是随机剪裁，并结合图像整体,更有效地评估图像美学质量。多次裁剪输入示意图如下：

102

人工智能篇



在美感度模型的训练框架里，多使用分类的方法进行最后评分的决策。

采用分类的方法，一般最后接一层 softmax，根据监督信息的不同，可以是判断图像好看与否的二元分类，或者是对多人评价分布的拟合。前者将输出好看的概率作为美感度分数，后者将十个等级的评分求加全和得到最终结果。

Google[ 3 ]提出 NIMA，通过学习每张图像的评分直方图，对任意给定的图像预测评级分布。

该方法能够更准确的预测出人类的偏好。

不同于以往的二分类处理，Kong 等[ 4 ]提出对图像美学进行排序，构造了新的数据集，对每张图片的 8 个美学指标进行打分。通过对每个采样图像对的排序学习建立基础美学特征提取器，并添加属性预测等内容分支，使用多任务学习方式进行微调。

1.1.2 应用

在实际应用中，我们用基于 ImageNet 预训练的 Resnet 作为特征提取基础模型。在网络输入上，我们有选择地采用多次裁剪得到的多个图像块，以及整张图像填充（padding）后缩放到固定尺寸作为输入，以兼顾全局信息与局部特征。

数据增广的处理上，常用的数据增广方法难以应用在该问题中，剪裁会对图像构图产生负面影响，缩放产生的形变影响了实际的比例和分辨率，因此我们只使用了水平方向的翻转。

值得注意的是，分数分布在中间档的数据通常处在模棱两可的状态，其数据量占比也比较大，采用二分类的方法，若强行设定一个阈值来区分低质量和高质量，会带来许多错误的标注。因此，在正负样例的划分上需要特别注意不要引入这些模糊数据。由于美感数据集获取不易，半监督、hard-mining 是优化模型的重要方法。

决策部分我们分别尝试二分类与 1-10 评分分布直方图学习两种决策方式。

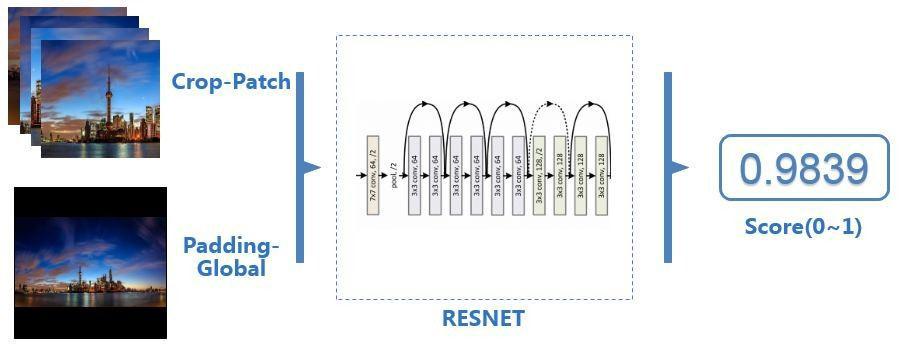
103

人工智能篇



由于评分分布需要大量人工标注，难以迁移到实际场景的数据中，我们只用该方法对美学数据集进行了训练，以提升特征提取的效果。

基于上述基础模型，我们又加入了自己标注的实际场景数据，替换最后决策层为二分类softmax，进行微调。最终将好看的概率作为美感度分数。美感评分网络示意图如下：



**1.2 可视化**

通过对图像的美感度打分，我们可以量化图像美感这一抽象概念。但是这些方法并不能直观的说明为什么一张图被判断为好看、不好看，以及是哪些方面影响了这张图像的美感。

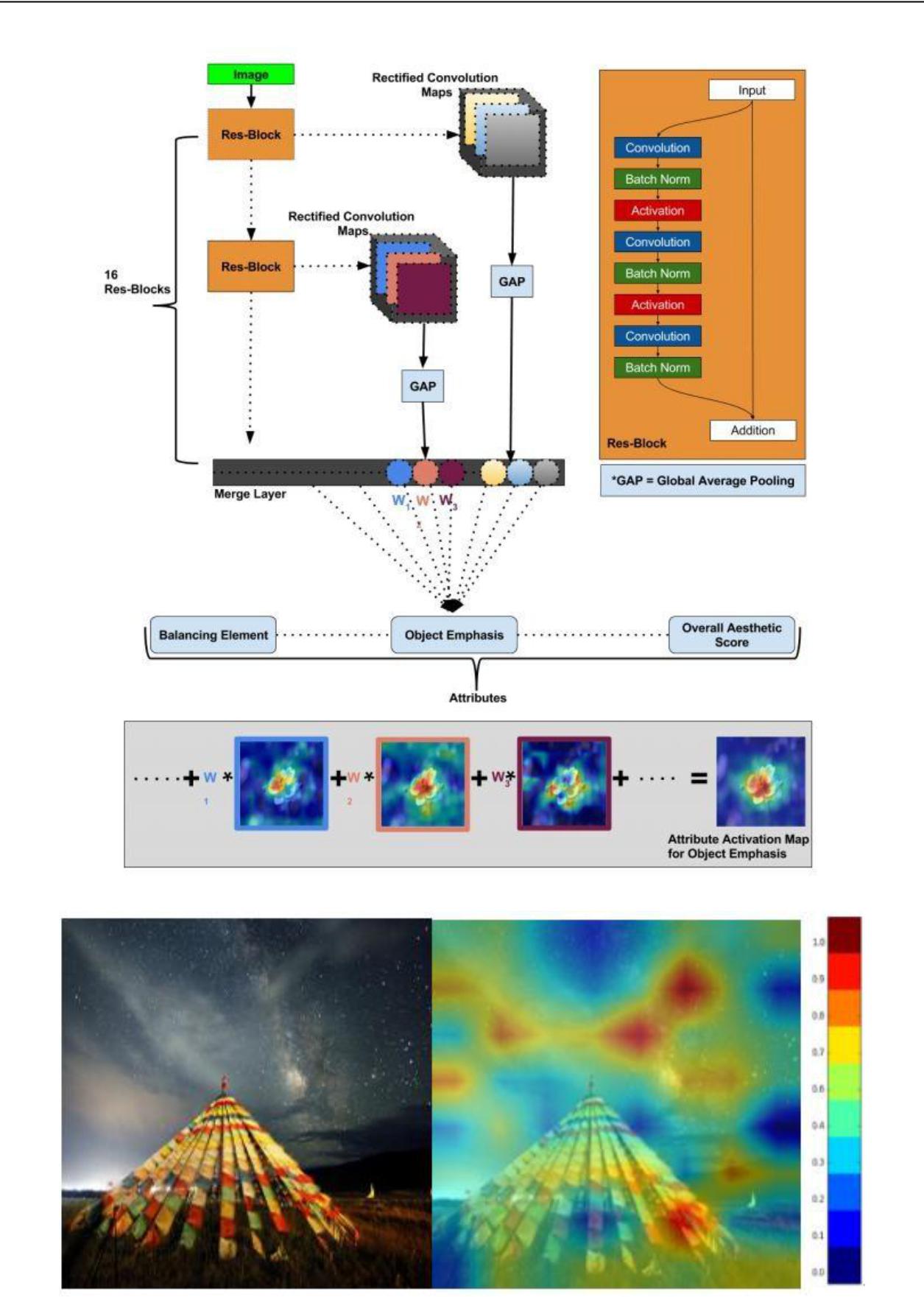
为了解释网络对图像美感的决策，我们尝试了 CAM 等可视化方法。

CAM 或梯度加权类激活映射（Grad-CAM）这类方法，都是通过得到每对特征图对应的权重，最后求一个加权和，通过对高层特征图的权重加权计算，可以达到很高的类别判别能力。这类方法使用流入 CNN 最后一层卷积层的分类权重或梯度信息来理解每个神经元对于目标决定的重要性，解释网络可能做出的决策。[5][6][7]

可视化方法与效果示意图如下：

104

人工智能篇



**1.3 图像内容匹配**

实际业务场景中，数据种类众多，在候选图集中还有许多图像不适合作为首图进行展示，比如表演时刻表、景区美食等不符合景区主题的图像。

为了选出更优质的图像，我们在美感评分基础上，进行了图像内容筛选的后处理。由于后处

105

人工智能篇



理的类别并非互斥关系，我们建立了一个多标签的分类训练模型，为每张图片打上相应的类别标签，只选取符合要求的类别图片，在其中使用高分图片最为最终首图。筛选最优质图片示意图如下：



**二、小结**

深度学习在图像识别领域取得了很好的成果，利用 DCNN 能够有效提取图像的特征，描述图片的美感。图像的美感问题是个高度抽象的问题，数据量相对较少的情况下，需要更深且权值级别更低的网络。为了得到较为全面的图像美学特征，图片全局和局部都有体现美感的特征，需要结合两者，才能更有效的对图像质量进行评估。

**相关文献**

1. Lu, Xin, et al. "Rapid: Ratingpictorial aesthetics using deep learning." Proceedings of the 22ndACM international conference on Multimedia. ACM, 2014.
2. Ma, Shuang, Jing Liu, and Chang Wen Chen."A-lamp: Adaptive layout-aware multi-patch deep convolutional neuralnetwork for photo aesthetic assessment." Proc. IEEE Conf. Comput.Vis. Pattern Recognit.(CVPR). 2017.
3. Talebi, Hossein, and Peyman Milanfar."Nima: Neural image assessment." IEEE Transactions on ImageProcessing 27.8 (2018): 3998-4011.
4. Kong, Shu, et al. "Photo aestheticsranking network with attributes and content adaptation." EuropeanConference on Computer Vision. Springer, Cham, 2016.
5. Zhou, Bolei, et al. "Learning deepfeatures for discriminative localization." Proceedings of the IEEEConference on Computer Vision and Pattern Recognition. 2016.
6. Selvaraju,Ramprasaath R., et al. "Grad-CAM: Visual Explanations from Deep Networksvia Gradient-Based Localization." ICCV. 2017.
7. Malu, Gautam, Raju S. Bapi, and BipinIndurkhya. "Learning Photography Aesthetics with Deep CNNs." arXivpreprint arXiv:1707.03981 (2017).

106

人工智能篇



**全球顶级算法赛事 Top5 选手，跟你聊聊推荐系统领域的“战斗机”**

**[作者简介]**朱麟，携程酒店研发部排序算法组资深算法工程师，主要负责携程酒店排序相关的 AI 项目，多年行业相关经验。博士毕业于中国科技大学，专注于推荐系统算法的应用和研发。

**摘要**

随着人工智能和大数据技术的飞速发展，推荐系统近年来非常流行，应用于各行各业。推荐的对象包括：电影、音乐、新闻、书籍、学术论文、搜索查询、分众分类、以及其他产品。

推荐系统产生推荐列表的方式通常有两种：协同过滤和基于内容的推荐，近年来很多基于这些方法的创新层出不穷。

携程酒店研发部排序推荐算法团队也在该领域的实践中作了很多有意义的探索，并在业务中得到应用。带着这些积累，我们在今年参加了一些关于推荐系统的国际知名数据挖掘竞赛， 2018 ACM WSDM 挑战赛和 2018 ACM RecSys 挑战赛, 均取得 Top5 的好成绩。

以下将结合两次比赛的实际内容，谈谈我们所采用的算法策略和心得。

**一、比赛简介**

2018 年 ACM WSDM 竞赛[1]，作为网络数据挖掘顶级学术会议 WSDM 的一部分，由 ACM 和亚洲顶级的音乐流媒体服商 KKBOX 联合举办。 比赛目标是搭建一个推荐系统，通过预测在一定时间内用户再次点播历史收听过歌曲给用户进行推荐，竞赛使用的数据集由音乐流媒体平台 KKBOX 提供，包含以下信息：用户、歌曲的 metadata，听歌活动与 App 的信息等等。

2018 年 ACM RecSys 竞赛[2]，作为推荐系统顶级会议 RecSys 的一部分，由 ACM 和国际知名音乐服务商 Spotify 联合举办。比赛的目标是搭建一个推荐系统，自动延续用户歌单。竞赛所用的数据集由 Spotify 提供，包含以下信息：用户、歌曲的元数据,，一百万个用户自建的歌单以及这些歌单的元数据。

**二、方法创新**

**2.1 对于特征工程的创新**

坊间常说：“数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已”。可见特征工程在机器学习过程占有非常重要的地位。

107

人工智能篇



在 ACM WSDM 竞赛中，我们在特征工程上做了充分的探索 ，除了构造常规的类别和统计特征，值得一提的是我们挖掘出了蕴藏在数据中的时序信息。考虑到给定的数据集是按出现的时间顺序排序的，所以尽管没有给明显的时间戳，我们依然可以从两方面探索考虑到输入数据的时间序列特性的特征：

2.1.1 物品的“年龄”相关特征

不像通常推荐比赛的设定，物品集和用户集都被假设是固定的，在 KKBOX 提供的为期两年的数据集（2016 和 2017）中，新发的歌曲被不断地加入到音乐库，新的用户也不断地使用KKBOX，可以理解为新的物品和用户不断加入到已有推荐系统。

一方面，推荐近期新发歌曲是非常重要的，因为在实际中用户往往偏好新的内容；另一方面，考虑歌曲的“年龄”和用户的“年龄”对偏差纠正也是很重要的，这是因为大量的物品和用户在数据集中不是“均一”分布的。

基于这些观测，我们首先构造了一些特征来衡量对应的客户/歌曲/艺术家/作曲家/谱曲家在此次用户-歌曲听歌活动前的出现时间。相似地，我们也构造了一些特征，表示对应客户/歌曲/艺术家/作曲家/谱曲家在此次用户-歌曲听歌活动后出现频率。

2.1.2 “会话”特征

对于在线流系统的推荐任务而言，通常从基于会话的角度来考虑用户和系统/物品更为有益。

会话是一组发生在给定时间间隔内的相互联系。

举个例子，一个用户可能在较短的时期内，会不时地听属于同一个艺术家或者同一个风格的音乐，识别这样的会话很有可能提高推荐准确率。

之前提到过，在此次竞赛中，听歌活动的时间戳并没有被直接给出，因此取而代之的是，我们贪心地将临近的属于同一个用户的听歌记录归为一个会话。

基于估计的会话，我们构造了三种特征，分别衡量用户在数据集中参与会话的数量，用户在一个会话中平均听歌数量，当前会话含有的歌曲数量。

在 ACM RecSys 中，我们考虑到出现在同一歌单的歌曲共现的性质，借用 word2vec 的思路，将歌单当作句子，将歌曲当作单词进行训练，得到每一首歌的 embedding，并基于这些 embedding 计算歌曲间相似度作为特征。

**2.2 对于模型的创新**

在实际的推荐系统应用中，各种机器学习方法百家齐放。能够基于已有方法，针对不同的实际问题作出自己的创新，集百家所长，往往能带来更优异的结果。

2.2.1 解决冷启动问题

108

人工智能篇



冷启动问题是协同过滤推荐算法中被广泛关注的一个经典问题，常常为实际推荐系统带来严重的挑战，它的存在严重影响了推荐系统的推荐质量。对于电子商务推荐系统，每天都有大量的新用户访问系统，每天都有相当数量的新项目添加到系统中冷启动。

在 ACM WSDM 竞赛中，我们亦遇到了同样的挑战。

在模型里，客户和歌曲都被表示成类别特征，所以此环境下的冷启动问题可以被理解成带有高基数的类别特征的“维度诅咒”：因为训练集的有限，不太可能观察到每个这样类别特征的可能的值，所以，如果学习模型过于依赖这些不大可靠的类别特征提供的信息，特征可能不能很好地推广到未来的测试集上。

在本次竞赛中，我们尝试通过借助去噪自编码和 dropout 的思想来改善这种问题，并且在没有高基数类别特征，像用户 id，歌曲 id，艺术家名字，作曲家，谱曲家等等的情况下重新训练模型。在这样去掉不可靠特征的情况下训练出的模型，和原模型融合后会得到更好的实验表现。

2.2.2 创新协同过滤

作为推荐系统经典方法的协同过滤可以分为基于用户和基于物品两种。

基于物品的协同过滤简单来说是利用某兴趣相投的群体的喜好来推荐用户感兴趣的信息，应用很广。但是，基于物品的协同过滤也存在缺点，往往会更倾向推荐热门的物品，使得一些小众的物品得不到重视而较少被推荐。

在 2018 ACM RecSys Challenge 中，我们针对基于物品的协同过滤进行一些改进，得到了较为不错的模型效果。

该竞赛的挑战目标，是开发出一个自动延续歌单系统。比赛提供的数据集可以简要概括成两部分，第一部分是百万歌单数据集（MPD）, 包含一百万个由用户创建的歌单和相关的元数据，比如歌单的名字，描述，艺术家/专辑/歌曲的数量等多种统计信息。

第二部分是一万个未完成的歌单和相关元数据， 分别含有 1-250 首歌不等歌单。评测指标是对该一万个未完成的歌单，从比赛给定的一百万首歌曲预测出最可能在该歌单的 500 首歌曲。

我们推荐的核心思想是假设给歌单 u 自动延续（推荐），将和歌单每首歌曲平均相似度最高的歌曲选出，来自动延续歌单。如何衡量这些歌曲和歌单的相似度？

一个简单直接的方法是，计算歌曲和歌单每首歌曲平均相似度。在参考一些以前的工作后，我们将该计算分为了以下三步：

1. 计算每首歌的特征向量

109

人工智能篇



对于每首歌曲，我们都可以得到它的特征向量, 已知数据集中有一百万的歌单，那么代表每首歌的特征向量将为一百万维，每一维的计算可见以下公式：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | *T* *u*  | | | *T* *v* | | |  |
|  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | *T* *v* | |  | ** |  |
| ( *X i* )[ *v*] |   | | | |  |  |  |
|  |  | | | |  |  | 0 |  |  |  |
|  |  |  |  |  |
|  |  | | | | | |  |  |  |



1. *v else*

T(u) - 歌单 u 的歌曲集合

T(v) – 对于任何包含歌曲 i 的歌单的歌曲集合

* + 0,1 - 超参，控制着长歌单的影响

1. 计算歌曲 i 和歌曲 j 的相似度

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | ( *x* )*T* *x* | | | | | *j* | | |
| *si* , *j*  |  |  |  |  | *i* | | |
|  |  | *P* (*i* ) |  | ** |  | *p* ( *j*) | |  | 1** |
|  |  |  |  |

P(i) – 含有歌曲 i 的歌单集合

P(j) – 含有歌曲 j 的歌单集合

* + 0,1 - 超参，控制着热门歌曲的影响

1. 歌曲 i 和歌单 u 的相似度

通过(a)和(b)计算得到歌曲与歌曲之间的相似度，对于一个未完成的歌单 u，我们计算歌曲 i 和歌单 u 中每一首歌曲的相似度，加权平均分即为该歌曲与歌单 u 的相似度。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *rui*  |  | 1 | |  |  |  *sij* |
|  |  | *T* (*u*) |  |  |
|  |  |
|  |  |  |  |  | *j**T* *u*  |

通过(a)和(b)计算得到歌曲与歌曲之间的相似度，对于一个未完成的歌单 u，我们计算歌曲 i 和歌单 u 中每一首歌曲的相似度，加权平均分即为该歌曲与歌单 u 的相似度。

至此，我们实现了基于物品的协同过滤推荐，通过超参α和超参β得以控制当歌单较长以及物品过于热门的影响，但是该方法步骤(c)中的加权平均计算歌曲与歌单相似度过于平等得看待所有的特征，使得相对的特征重要性被忽略了只能得到次优解。

针对以上问题，我们提出判别式重新加权方法（Discriminative Reweighting）进一步改进。

2.2.3 判别式重新加权

我们采用的判别式重新加权算法主要借鉴了 SLIM 算法。

110

人工智能篇



SLIM(Sparse Linear Model) [3]，中文名是稀疏线性推荐算法，该方法基于物品相似度的推广形式， 以 M 表示评分矩阵，S 表示相似度矩阵，在计算评分时作为对应物品的权重，优化目标即为使得 M 和 MS 差值最小，同时通过添加对权重的正则使得权重更加稀疏以此达到更好的模型推荐效果。

相似 SLIM 模型，对于以上平均看待权重而忽略特征重要性的问题，通过求解以下 L2 正则 SVC 问题，我们可以有效地学习到更多具有判别意义的权重，从而学得更精准的歌曲和歌单的相似度：

|  |  |  |
| --- | --- | --- |
| min | 1 | *wT w*  *C* (max(1 *yi wT sui* , 0))2 |
| 2 |
| *w* | *i**T* |

其中标签 *yi* 表示 *i* 是否属于 *T* *u* 。

2.2.4 判别式重排序与模型集成

如今推荐系统领域，乃至机器学习各领域，常常用多种不同类别的模型集成来完成推荐，比如 Facebook 利用 XGBoost 和 LogisticRegression 集成，得到的模型在点击率得以极大的提升。多模型集成不仅能充分挖掘数据特性，也能更好综合预测结果，提升模型表现。

在 ACM RecSys 挑战赛中，考虑到协同过滤方法仅仅依靠歌曲和歌曲，歌曲和歌单共同出现的频率来计算相似度，但还有其他数据，像歌曲名，歌单名等等的信息我们尚未使用，因此我们集成了原有的协同过滤模型和 GDBT，将协同过滤做粗排，筛选出初步的候选集，然后通过元数据构造特征，与协同过滤计算的概率分特征一起，通过 GDBT 进一步作精排，得到最终的推荐结果。

在 ACM WSDM Cup 中，通过嵌入式模型，如 Factorization Machine[4]或深度神经网络，将类别 id 嵌入低维度空间来表示潜在的偏好，这种方法能推广到先前未观测到的类别特征对，因此除了基于特征工程的分类模型-Gradient Descent Boosting Tree(GDBT)，我们还采用了 factorization machine 来学习每个用户歌曲对的潜在因素，并基于观测到的似然值而排序给定的歌曲。

**三、总结**

在推荐系统领域，通过对数据集不断挖掘更为有效的特征，针对已有方法在特定问题上的创新，以及更充分有效挖掘数据特性的模型集成往往能带来更加优质的推荐，带来更好效益的模型，更充分发挥人工智能的优势去服务广大的消费者。

在携程的日常服务中，个性化的推荐算法以及不断提升的推荐质量，也将为旅行者带去更良好的消费体验，找到让每一位旅行者都满意的旅行产品。

（携程酒店研发部陈毅鸿，何博文对本文亦有贡献）

111

人工智能篇



**参考文献**

1. Y.Chen, X. Xie, S.-D. Lin, and A. Chiu, "WSDM Cup 2018: Music Recommendationand Churn Prediction," in WSDM,Marina Del Rey, CA, USA, 2018, pp. 8-9.
2. C.-W. Chen, P. Lamere, M. Schedl, and H.Zamani, "Recsys challenge 2018: automatic music playlistcontinuation," in RecSys, 2018,pp. 527-528.
3. X. Ning and G. Karypis, "Slim: Sparselinear methods for top-n recommender systems," in ICDM, 2011, pp. 497-506.
4. S. Rendle, "Factorization machines,"in ICDM, 2010, pp. 995-1000.

112

人工智能篇



**携程实时智能异常检测平台的算法及工程实现**

**[作者简介]**陈剑明，携程网站运营中心数据分析高级经理，负责网站容量规划、ATP基线预测及 RCA 损失计算、成本分摊、运维数据仓库建设，利用机器学习和深度学习相结合，进行运维方向的数据分析与预测。本文来自陈剑明在“[2018 携程技术峰会](https://mp.weixin.qq.com/s/8tGK_gQH3VqL9K2OTwgxYw)”上的分享。

在运维领域，异常检测是很重要的基础，决定了告警的数量和质量，也影响了发现异常之后所需要采取的行动的可靠性。

然而实际生产环境下异常的判断往往带有主观性，准确率、召回率等指标缺乏客观依据，标注成本又高，导致误报和漏报始终都是一对矛盾的存在。

本文主要介绍我们在这些已知问题基础上，进一步优化“异常检测”算法的探索和成果，并介绍了如何解决告警实时性的工程技术方案。

**一、引言**

日常工作中我们经常会接收到频繁的异常告警，处理起来眼花缭乱，容易遗漏问题点。如何降低误报率，让有限的注意力集中在真正需要关注的异常上？

在机器学习领域异常检测是个很大的主题，传统的统计学方法就有很多，深度学习领域也有不同的算法模型来检测异常，很多论文都对此做过研究和探索，然而直接借鉴并不能让我们获得理想的效果，需要一种方法可以帮助我们：

* 降低报警总量到可以人工逐个处理的程度
* 不能以增加漏报真正的故障为代价
* 提升告警的实时性
* 算法即服务，有较强的可移植性

**二、大而全的监控衍生出的问题**

不管运维还是开发，大家都明白一个道理，系统跑得好不好，监控工具少不了，监控是我们的眼睛。

特别是规模比较大的系统，我们需要一个大而全的实时监控系统，如果这个系统还能判断业务异常，并能实时发送给相应的人员来处理，那就更棒了。

DevOPS 火了这么多年，相信很多同仁也在自己的公司实施部署了具备这样能力的监控系统，我们想和大家探讨的是，接下来可能会面临一些什么问题，以及我们一路是怎么走过来的。

113

人工智能篇



当我们有了一个这么强大的实时监控告警系统，将几千上万数十万个监控指标接入进去的时候，问题就来了，这么多指标该如何去设置告警？

不那么重要的指标可以不设告警，等出问题的时候再来追踪查看；有些系统级的指标可以设置简单的告警规则，比如 CPU 使用率、磁盘利用率，超过一定百分比报出来就好；还有很多指标不那么好设置统一的规则，比如访问量，响应时间，连接数，错误量，下单量，怎么样才算是不正常？这在很多时候是个感性的判断。

为了让自己变得理性，我们自然而然地会去量化，常见的量化方式是和过去的某一个时间相比较，比如和一周前的同一时刻前后几分钟的均值相比，降幅如果超过了一定比例，并且连续出现，就判断为异常。这种方式面临两个局限：

1、成千上万个指标，需要人工去设置，费事费力，业务变化会导致数据形态上的变化，规则维护成本很高；

2、每个指标的业务背景不同，降幅超过多少才算是异常，这是个艺术，没有标准答案，每个人都有自己的判断，需要经验。

如果监控系统是我们的眼睛，那么报警系统就是神经。面对这么多监控数据，怎么样让神经既不那么敏感，也不那么大条，还能自动适应各种刺激，这是我们尝试去探索和改善的问题。很自然的，我们希望引入一套算法来解决这个问题。

**三、统计模型的困扰**

一旦引入算法，在整个建模过程中，绕不开两个问题：明确的算法评估标准和足够的样本数量。

首先，明确的评估标准，在算法迭代和检验阶段非常重要，否则算法调优就没了方向。虽然二分类问题在理论上有召回率和准确率这些评价指标，但这两个指标在我们这种检测场景下本身不可衡量。

原因是不管监控如何强大，总有我们发现不了的异常，而一个人认为的异常，换一个人来看，也许就不认为是异常。异常本身没有明确的定义，也没有一个全量的边界，如何检验一个算法的好坏？

其次，生产系统中能够确定的异常，相对于监控的数据总量来说，样本非常的少，需要大量标注，标注又是个体力活，况且在这个场景下，标注人员的标准都不一致，而且这些少量的样本还分布在这么多的指标里，我们如果要从这么少的样本里学习出一些特征，也将是非常困难的。

那怎么办？

面对这两个近乎无解的问题，我们没有什么特别好的方法，但希望能找到一个子集，在这个

114

人工智能篇



子集阈内这两个问题是有确定边界的，然后寻求一个最优解，再慢慢拓展到整个数据集合。

首先是对算法模型的评估标准，我们需要知道哪些异常，或者说故障，是大家一致认可没有歧义的。

我们有个 NOC 团队负责 7\*24 小时接受来自各个渠道的告警和报障，包括监控系统、用户报障、内部人员发现的问题，如果经确认是一个影响了订单的生产故障，则会记录下来，这就给第一个问题提供了一个相对最为接近可衡量状态的条件。

首先一个故障只要影响了订单，那么对于订单这个指标来说，这是无可争议的异常，其次，所有问题都汇集到一个处理中心，我们可以大致认为，这里所记录的问题，就代表了系统存在的所有可见异常。

所以，我们把目光锁定在所有订单类型的监控项上，一是因为只有这类监控的异常才是会被广泛认可，二是这类问题只要发现就不会被错过，一定会有人确认是不是一个真的异常。这就为我们的算法迭代和优化提供了一个明确的检验标准和可靠标注。

**四、算法选择和设计目标**

我们以订单这类指标为入手点来调试算法，接下来就是算法的选择。

不论采用哪种算法来检测，一个显而易见的好处就是可以减少规则的维护成本，我们不再需去给每一个指标设定合适的告警规则。

目前业界采用比较多的方式是引入统计分析的各种方法，框定一个滑动的样本集，对这个样本集进行一些数据处理和转化，经过归一化，去周期，去趋势，再将最新采集到的数据点经过同样的转换，和样本集的残差序列的统计量进行比较，比如距离、方差、移动平均、分位数等，超出一定的范围就判断为异常，或是综合各种离群点计算的方法来做个投票，多数算法认为异常则报异常。

起初我们也借鉴了这种做法，却发现虽然可以不用维护告警规则了，但报警的质量并没有提升。

和规则化告警面临的问题是类似的，当我们把置信度设成一个理论上的显著值，比如 95%时，会检测到很多的异常，当我们不胜其烦，把置信度调高时，报出的异常总量是会不断减少，但与此同时，会发现相比于原来的规则化告警，遗漏没有报出来的故障也逐渐增多，误报和漏报始终都是一对矛盾在出现在天平的两端。

通常的思路是，当两个因素互斥的时候，我们需要寻找一个折中点，让两者达到均衡以获取一个相对最稳妥的结果。

但生产环境不接受稳妥，宁可报警多一些，也不能接受漏报。但另一方面，报警接收人员要开始神经衰弱了，每个都仔细排查没那么多时间，万一漏看一个刚好和故障相关，又得担责任，他们想收到真正值得排查的告警。现实逼得我们不能搞平衡，必须鱼和熊掌要兼得，这

115

人工智能篇



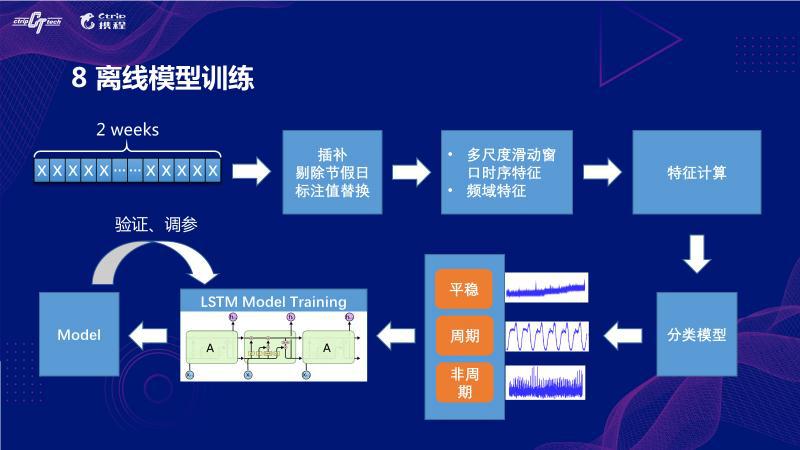
并不容易。

我们需要设计一套新的算法，降低报警总量到可以人工逐个处理的程度，同时不能以增加漏报真正的生产订单故障为代价，并且这套算法的设计还不能太复杂，影响到告警的实时性，最好还能做到算法即服务，有较强的可移植性，提供给其他的监控系统使用。

自然而然的，基于神经网络的深度学习算法成为我们进一步探索的工具。

RNN 模型比较适合处理序列变化的数据，符合我们时序特征的场景，而他的改进版 LSTM 模型，能够通过控制传输状态来选择性地记住较重要的长期数据，能在更长的序列上有良好的表现，业界也有很多成功的应用。这里重点介绍一下如何引入到我们的场景中。

**五、算法的描述和检验**

****

这是一个离线训练的过程示意图。

我们把历史数据拿过来，先做个清洗工作，对缺失值进行插补以及节假日数据的剔除。剔除节假日是因为训练当中如果包含了这部分数据，模型就会有偏差。当然缺失值如果超过 20%，那就干脆不训练。

然后对这个序列做特征提取，特征工程的目标是把时序序列分为三大类，周期型、平稳型、非周期。

我们使用了多尺度滑动窗口时序特征的方法，将一个滑动窗口内的数据和前 n 个周期做统计量上的对比，均值、方差、变化率等这些，这样基本上就可以把明显的周期性和平稳型数据给分离出来。

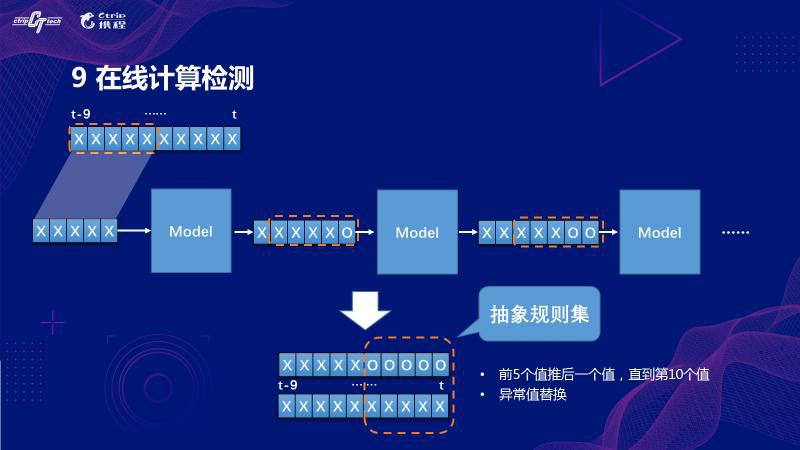
116

人工智能篇



剩下的时序中，有些是波动很大的随机序列，有的则是带有趋势的周期性序列，通过时序分析法把周期性去掉，再用频域分析尝试分解成频谱。对于带有明显频谱的，则归类为周期型时序，而频谱杂乱的，则归类为非周期性。

为什么要分成三类后面会讲到。分出来之后我们定义 LSTM 需要的各个变量，然后是调用 TensorFlow 进行 LSTM 模型训练、验证和调参的过程。



在线计算检测阶段，滑动窗口取最近的 10 个数据点，用前 5 个点作为模型的输入来预测后 1 个点的值，循环输入模型直到预测出后 5 个点的值，并用这几个预测数据点和实际值进行比较。

除非遇到极端情况，否则只一个点是无法判断是否异常的，所以我们需要推测 5 个点，不是将来的 5 个点，而是过去的 5 个点，结合一些基本的规则，来对这 5 个点的实际数据做出异常判断。

这里还是需要结合最基本的规则，而什么样的数据类型采用哪些基本的规则，经过反复尝试发现是不同的，这就是为什么我们要先将数据指标分类。

我们将异常分为连续 1 个点、2 个点、3 个点这几种情况，每种情况下抽象出来一些最基本的规则，连续几个异常点，结合不同的变化幅度，临近周期内的统计量变化，以及多个点位的形态变化，设定了数套规则集，对应了高中低 3 个不同的检验敏感度，并调校验证不同敏感度下模型的表现。

117

人工智能篇



我们以两个指标作为算法迭代和优化的依据：

1、和现有的规则化系统相比，同样的监控指标下，算法产生的告警量是否少于当前规则系统所产生的告警量；

2、以所有汇集到 NOC 的生产故障为全集，通过算法检出的数量是否高于当前规则系统检出的数量。

最终得到这样一个结果： 相比于规则化告警系统，算法的告警量平均压缩了 10 倍左右，而检出的故障数量反而还略高。

虽然故障检出率提升了，但还是存在漏报的情况，我们分析下这些漏报的故障，主要表现为以下几个方面：

* 指标看山去正常波动，肉眼及系统均无法检出（用户人工报障）
* 业务订单量较小导致漏报—— 绝对值越小，少量波动就会造成巨大的影响
* 历史数据波动比较剧烈，异常下跌幅度小—— 数据随机性太强，不容易准确分辨
* 趋势变化明显—— 缩短模型训练周期捕捉周期性

**六、实时性工程**

尽管如此，这已经是一个让各方都能相对接受的结果，而需要解决的最后一个问题就是实时性。

我们原本的算法都是 python 的实现，在实时处理上 python 没有什么优势，我们把采集到的数据落地，清洗插补，过模型计算，再触发报警，这些没有办法流式处理，只能通过 cron 的方式一步步调用。

118

人工智能篇

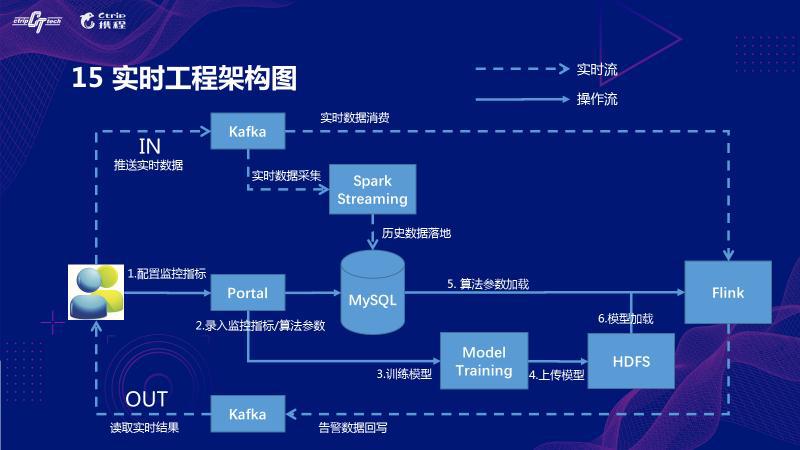


而大家知道 cron 的最小时间间隔是 1 分钟，这就导致了从收到异常数据点到发出报警信息，中间需要经历 3~4 分钟的时间间隔，而一个规则告警系统，数据落地之后只需要对比规则，只需要 1~2 分钟时间就能将告警发出来，对于重要的业务指标来说，这两三分钟的报警延迟就意味着真金白银的损失，也是不可接受的。

实时化工程的方案选型，我们考虑了 storm/spark streaming/Flink 这几种，最终选择了 Flink，原因在于它能满足我们的要求，滑动窗口灵活，数据可以基于自身的时间戳来统计，不会因为数据延迟而落到下一个时间窗口来统计，这会给我们减少很多数据处理的麻烦事儿。加上 Flink 本身是为实时计算设计的，容错性比较好，我们不用考虑很多数据达到的异常问题。同时也预留了支持秒级数据采集粒度的能力。

但是有一个问题就是检测结果的数据校验需要实时计算，否则来回调用 python 会增加时间消耗，所以就将算法以 Java 代码重新实现。

我们的系统设计是这样的：



用户将每个指标的实时数据按照一定格式推送到 Kafka 队列中，并且通过 Portal 确定哪些指标是需要做异常检测的，如果指标有历史数据的话，提供 2 周的历史数据用于训练模型，或者可以不提供，等待两周的数据积累。

我们拿到这些数据之后，对所有满足训练条件的指标（有足够的历史数据）进行离线训练，生成模型之后放在 HDFS 中，Flink 加载新生成的模型，每个流过的指标如果有匹配的模型，则流入模型计算，否则丢掉，最后将计算结果回吐到指定的 Kafka 队列，供用户方消费。所有模型每两周重新训练一次，若发现用户上传新的指标，则触发训练，Flink 每 5 分钟检查一次最新的模型并加载替换老模型。

119

人工智能篇



这样一来，实时数据不需要落地就能进入模型马上计算得出结果，相比于规则告警系统先落地再计算的方式平均提升了 40s 左右，更接近实时。而这种方式也将算法和工程实现抽象出来，对外以队列的方式提供输入输出，任何一套监控系统只要按照约定的格式传入时序数值，就能使用这套方案来进行实时检测。

**七、算法的局限**

这就是我们目前为止的尝试，大家可以发现，这种方式还是存在不少局限性的，比如每一个指标训练一个模型，每个模型十几兆，说实话不小，每次训练需要 10 分钟，所有模型全量训练一次，对服务器的压力不小。

10000 个指标如果 CPU 资源不够的话，光训练就要好几天，运行时也需要 100G+内存来加载，所需的计算资源随着需要检测的指标数量呈线性增长关系。

其次对绝对值较小的指标没有好的解决方案，因为这类指标稍有变化，震荡幅度就特别大，很容易误报。还有就是非周期性随机序列，这种指标无法学习出什么模式，也就无法判断了。

**八、尾声**

我们现在仍然在不断尝试，希望能提取出一个通用模型，以在面对更大监控项的时候能节省算力。也非常希望能和这方面感兴趣的朋友们多多交流学习，路漫漫其修远，我们一直都在探索的路上，心存敬畏。

120

人工智能篇



**行业智能客服构建探索**

**[作者简介]**戴祥鹰，就职于携程数据智能部。此前先后供职于腾讯、百度，主要从事搜索、推荐、知识图谱、自动问答等相关工作。硕士毕业于哈尔滨工业大学。本文为作者加入携程前所做项目工作的经验总结。

**引文**

近年来科技产业蓬勃发展。一方面，随着互联网的普及和发展，用户在使用互联网产品过程中产生了海量的数据；另一方面，硬件设备和算法也取得重大突破。

在数据积累、算法、算力都取得巨大进步的前提下，人工智能爆发；伴随着 AlphaGo 战胜人类冠军，这个概念也开始进入了普罗大众的视野。

谈到人工智能，我们首先要了解一个重要概念：图灵测试。

下面引用其在百度百科中的解释：图灵测试是指测试者与被测试者（一个人和一台机器）隔开的情况下，通过一些装置（如键盘）向被测试者随意提问。进行多次测试后，如果有超过30%的测试者不能确定出被测试者是人还是机器，那么这台机器就通过了测试，并被认为具有人类智能。

图灵测试在上世纪 50 年代提出，从图灵测试的解释中可以看到，人机对话系统是衡量人工智能的重要场景，也是随后人工智能研究的重点方向。

在本轮的人工智能热潮中，人机对话系统依然是重点方向之一，并且以智能客服或智能助手的方式落地，多数用以解决企业在线服务中人工服务成本高，响应速度受限和服务时间受限等业务问题。

本文重点聊一聊在医疗行业智能助手探索中遇到的问题，以及为此尝试的方法。把客服类项目中需要的对数据构建、用户问题分析及理解的思考过程分享给各位读者，希望对同类项目的思考有所帮助。

**一、问题背景**

我们面对的是一个在线医疗服务场景：患者在线上通过网站或者 app 提出问题，医生在线做出回答，服务的过程会产生多轮的问答交互。

在这个场景中，业务上有两个突出的问题；第一，在线医生资源不足；第二，医生响应回复不及时。这两个问题影响用户的产品体验，平台信任度，进而影响用户留存，用户转化等业务指标。

在这个背景下，我们提出利用 AI 技术，构建一个可对话的医疗智能助手，用来缓解以上业

121

人工智能篇



务问题。

最终，我们花费四个月的时间开发和迭代了一个智能助手，它可以在医生没有响应时给予用户及时的反馈，并通过与用户对话来收集用户的信息，还会自动计算用户的高概率疾病，用于医生参考。（注：系统是由具有医学背景的客服人员使用，不会出现机器人直接回答用户问题的情况）

由于行业的特殊性，系统对技术指标的要求是非常严格的，即，在高精确度的前提下，尽最大可能提升召回率。

下面我们就来看看智能助手的整体思路。

**二、解决思路**

整体方案分为两部分：

1）行业物料构建；

2）智能助手搭建；

**2.1、行业物料构建**

2.1.1 数据获取

理想的情况是拿到标准的电子病历；一方面，从<用户提问,确诊疾病>数据对中学习疾病分类和预测模型，用于对用户的病情自述做科室和疾病预测；另一方面，从问诊记录中统计疾病与症状(含体征)的关联关系，并计算转移概率，从而可以在问诊过程中动态计算下一步需要问询的状态。

实际情况是，电子病历是医院、医疗机构的机密数据，我们无从获取，因此需要寻找此类数据的替代品。

从目标需求出发，我们最终锁定了两类公开的替代数据源。一类是在线医疗网站上的多轮问答数据，从这些数据中可以标注出问题与最终疾病的 pair 对。另一类是医学书籍，从中我们可以抽取整理出疾病-症状的关联关系。

122

人工智能篇



互联网上的医疗对话数据示例

2.1.2 数据处理

2.1.2.1 数据结构化

无论是对话数据还是书籍数据，都有多个来源和版本。我们首先将其转换为统一的数据格式，再从中抽取出关键字段信息，最后辅助专家标注审核，给数据附上标签信息。例如，对话数据中关于结论疾病的部分，需要医学专家审核确认，缺少的给予补充。

2.1.2.2 数据归一化

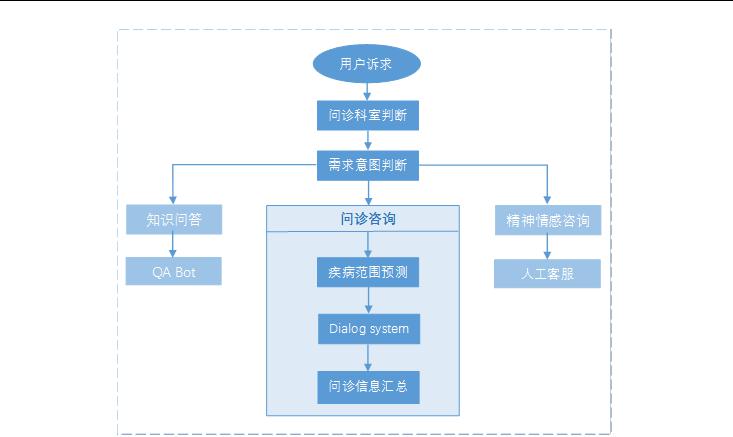
数据归一化重点针对核心医学概念，疾病及症状。由于语言描述的丰富性，一个概念通常会有多种表达方式，例如：一个疾病叫小儿腹泻病，又叫婴儿腹泻病，小儿消化功能紊乱等。我们参考医学系统命名法 SNOMED CT[1]，将其中一个选为标准名，其它作为别名，并建立映射关系。

**2.2、智能助手搭建**

2.2.1 整体框架

123

人工智能篇



智能助手相关的服务逻辑如上，出于业务敏感性考虑，隐去了部分模块和细节，整体流程简要概述如下。

在经过科室判断和用户意图识别后，被甄别为真正有多轮问诊诉求的用户被引至问诊服务，知识满足类需求由自动问答服务来满足，而精神/感情咨询类直接由人工服务。

问诊服务核心的功能是：通过与用户进行多轮问答，询问和收集用户的信息，并预测用户最可能的疾病范围。在与用户对话过程中，问诊助手提出的问题要符合一定的条件：1) 符合客观逻辑，如：不应该向男性患者询问妇科问题，不应该把仅适合儿童的问题提给成年人。2）使得对话过程尽快收敛，即每轮的提问应该在当前状态下最有利于疾病范围确定；或者最有利于确定 bot 无法满足。

下面，我们就问诊模块中几个关键任务点展开，阐述面对的问题和技术方案。

2.2.2 关键任务

本节介绍框架下的几个关键任务。

2.2.2.1 疾病范围预测

疾病有几万个之多，如果让问诊模块直接判断所有的用户问题，每次对话的搜索空间太大，而对话轮次必然是有限的，现实中 Bot 不可能跟用户询问太多轮次；因此，必须减小会话过程的搜索空间。

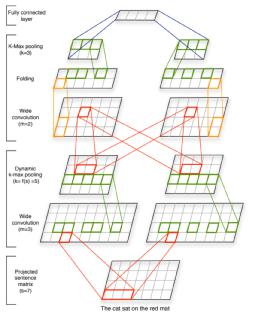
我们采用逐层分解的办法；第一层；科室划分；第二层，需求意图判断；第三层，静态疾病范围预测(发生在用户提问的第一轮)；第四层，动态疾病范围预测(发生在除第一轮后的每一轮回话中)；通过逐层划分，保证问诊模块每次会话时的搜索空间是可控的。第三层和第四层区别在于处理的输入特征不同，采用的模型是一致的。

124

人工智能篇



下面就介绍用于疾病范围预测的模型，我们将其定义为一个多分类问题，采用 Dynamic Convolutional Neural Network(DCNN)[2]模型来实现。



上图描述了 DCNN 算法的运行过程；其中，以长度为 7 的句子为例，embedding size 为 4；网络有 2 层卷积，卷积宽度分别为 3 和 2；卷积后的 k-max pooling 中 k 的取值分别为 5 和3。

DCNN 与一般 CNN 的区别在于，max-pooling 的维度取值是动态计算的，有利于特征提取；另一个区别是多了一个 feature folding 层，用于特征叠加。正因为该模型的特点决定了其可以更好地提取特征，符合业务场景中存在多个特征片段的特点，我们才选取其作为预测模型；从项目效果表现上看，DCCN 也超出了同层数 CNN 模型、及 FastText 模型的表现。

利用 DCNN 计算用户输入在科室下目标疾病范围上的概率分布，可以容易的得到 Top-k 个目标疾病，并通过医疗 Knowledge Graph 中症状-疾病间的关联关系排除部分非目标疾病，进一步缩小搜索范围。

2.2.2.2 对话过程中的信息抽取

对话过程中，智能助手需要不停的从用户的反馈中获取关键信息，例如：患者性别，患者年龄，过往病史等基本信息，以及最重要的症状表现；表现包括出现和未出现，在症状上来说就是有出现该症状或者没有出现该症状。

125

人工智能篇



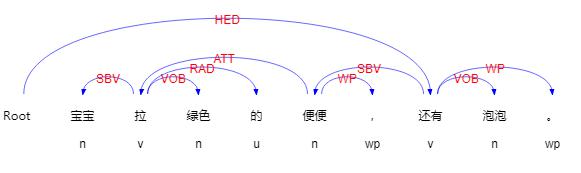
逐步积累的用户信息用于决策后续的搜索空间，因此，对话过程中的信息抽取成为另一个关键任务。信息抽取是一个非常有挑战的任务，鉴于业务对抽取结果的高要求，我们采取了 2 种方法组合的形式来解决片段抽取问题，采用分类模型来解决正负语义判定问题。

下面就介绍解决信息抽取的主要方法。

1）语义解析

通过对数据的分析我们发现，部分症状描述是由一定规律的，例如符合动宾关系，例如发烧，打喷嚏，拉肚子等。因此，我们通过句法分析获得句子结构，通过定义句法模板提取后续片段。

例如，对于用户的问题：“宝宝拉绿色的便便，还有泡泡。”，句法结构如下：



通过获取句法结构，可以抽取候选症状；例如，我们提取以 VOB 为核心的内容块，可以得到：拉绿色的便便、便便还有泡泡这样的症状描述；然后再对症状片段进行语义归一。

2）Bi-lstm+CRF

句法解析能解决一部分抽取问题，但会话中普遍存在口语化表达，导致基于句法解析的方式召回不全，因此我们必须寻找新的办法。

我们将症状抽取定义为一个标注任务，即从句子中识别什么位置是一个症状的开始，什么位置是症状的结束。下面是一个句子的标记，O 代表其他，B 代表症状开始，I 代表症状内部。

宝宝嗓子有痰，腹泻并伴有拉水的症状。……

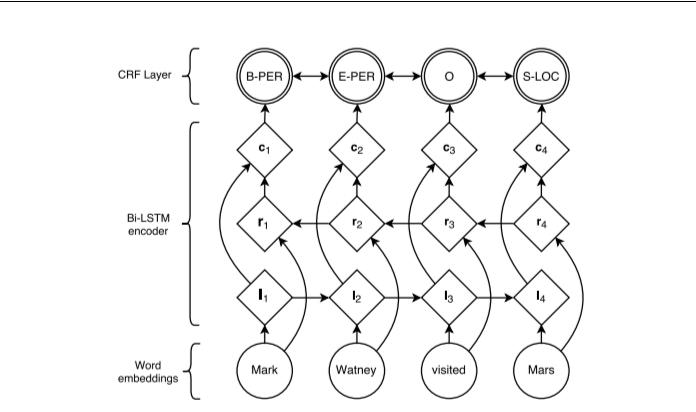
O O B I I I O B I O O O B I O O OO ……

在这个任务中，我们引入了 Bi-lstm+CRF 来解决序列标注任务，该模型将考虑双向的信息作为输入，使用 LSTM 提取序列特征，而 CRF 有效利用句子层面的标记信息；可以获取整体更好的标签序列。

下图是模型的示意图，引用于模型的经典论文[3]。

126

人工智能篇



2.2.2.3 问题生成

问题生成指，在对话系统经过计算确定下一个要提问的问题点后，系统生成自然语言问题，用于向用户提问。问题生成涉及两个点，一个是选择问题生成点（这也是对话管理的一部分），另一个问题是话术构建及拼接。下面简单介绍一下提问点计算和模板选择时的一些考量。

1）提问点计算

a、深度：症状-> 症状子属性

b、广度：该疾病下并列的其它症状

因为部分症状存在子属性，例如症状的轻重缓急，症状的持续时间等维度。某一症状被选取为下一个提问点时，需要考虑两个维度上的选择，选择的依据是根据历史数据计算那个维度更利于对话收敛。

例如，疾病“咳嗽变异性哮喘”的主要症状包含咳嗽，而且是夜间或凌晨咳嗽特别厉害，其它时间几乎没有咳嗽发生，那么当会话获取到该疾病的几个关键症状且包含咳嗽时，只有往前判断一步判断咳嗽发生时间，即可以大概率判断用户是否感染该疾病。

2）生成内容

从对话数据中总结话术模板，基于模板生成问题。在模板选择时，同类问题尽量随机选择候选模板，避免用户认为是跟机器人在交互。

**三、技术延伸**

在该技术系统实现的基础上，以医学对话系统构建为目标，在基础系统上引入强化学习技术，并在第三方标注的独立数据集上进行了实验和验证，效果与传统的方法相比获得明显提升，

127

人工智能篇



我们的成果在自然语言处理会议 ACL 2018 上发表，具体参见相关的论文[4]。

**引用：**

1. https://www.nlm.nih.gov/healthit/snomedct/
2. A Convolutional Neural Network for ModellingSentences http://www.aclweb.org/anthology/P14-1062
3. Neural Architectures for Named EntityRecognition； https://www.aclweb.org/anthology/N16-1030
4. Task-oriented Dialogue System forAutomatic Diagnosis http://www.aclweb.org/anthology/P18-2033