如何将深度学习与你正在做的事情相结合？

前言

人工智能是目前各行各业最火热的技术，如果说前两年是『互联网+』，那么这两年一定是『AI+』，利用深度学习的技术，给各行各业赋能，并提高效率，是企业智能化的一个方向。

从AI的结合方向来说，一般有2种，一种是行业结合，例如量化投资和智能投顾。

量化投资，量化投资就是利用计算机技术并且采用一定的数学模型去实践投资理念，实现投资策略的过程。它能严格执行投资策略，这样可以克服人性的弱点，如贪婪、恐惧、侥幸心理，也可以克服认知偏差。并且对海量数据的多角度观察，能快速跟踪市场，扩展投资机会。

在比如，AI与教育，国内的几家在线教育机构都有涉猎。英语流利说，用语音识别的方法，来判断用户的发音是否准确；乂学教育，将高中小学的题目，依据语义识别，将题目背后的知识点挖掘出来，用来诊断用户对一道题的不会做，究竟是哪些知识点不会，可能涉及到的小初高的各个知识点，分别列出，方便针对性的练习；小猿搜题利用OCR技术，获取题目的文本资料，最后对题目在数据库中及知识点中做匹配。

另一种是技术结合，这里的技术指的是互联网企业从业人员的各个角色，例如运维、开发、运营、搜索、推荐等。也是我们这里主要介绍的内容。这里可以做的事情很多，一般是一些重复性工作，有一定规律，但是这种规律不容易用规则描述，都可以尝试用深度学习的方法来赋能。

当然，深度学习或者机器学习在上面两种结合的情况下获得成功案例，离不开这几大要素：

- 数据
- 标注
- 工具（算法和系统）
- 应用场景

我们也逐个来展开。最后介绍深度学习中遇到的几个挑战，以及解决的思路。

一、如何将深度学习与你正在做的事情相结合

智能运维

运维的发展目前经历了从基于规则到基于学习的。运维面临的最大挑战就是：在互联网公司很难人工指定规则。在一个较大的企业中，它的特点如下：

规模大：

- 100多个产品线
- 上万个模块
- 几十万台服务器
- 百万级KPI监控

变化快：

- 每天上万个软件更新
- 互联网从业员工流动性强

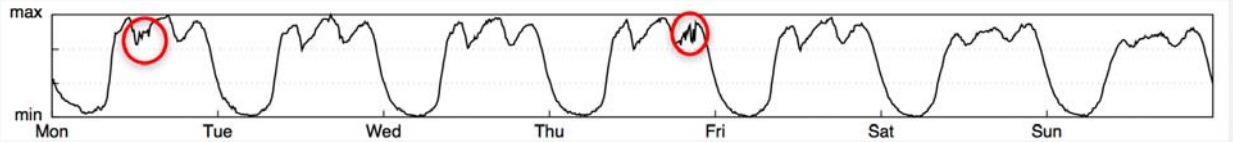
那么在运维中，都有哪些场景，可以尝试引入机器学习的算法呢？

场景一：事故的根因分析（RCA）

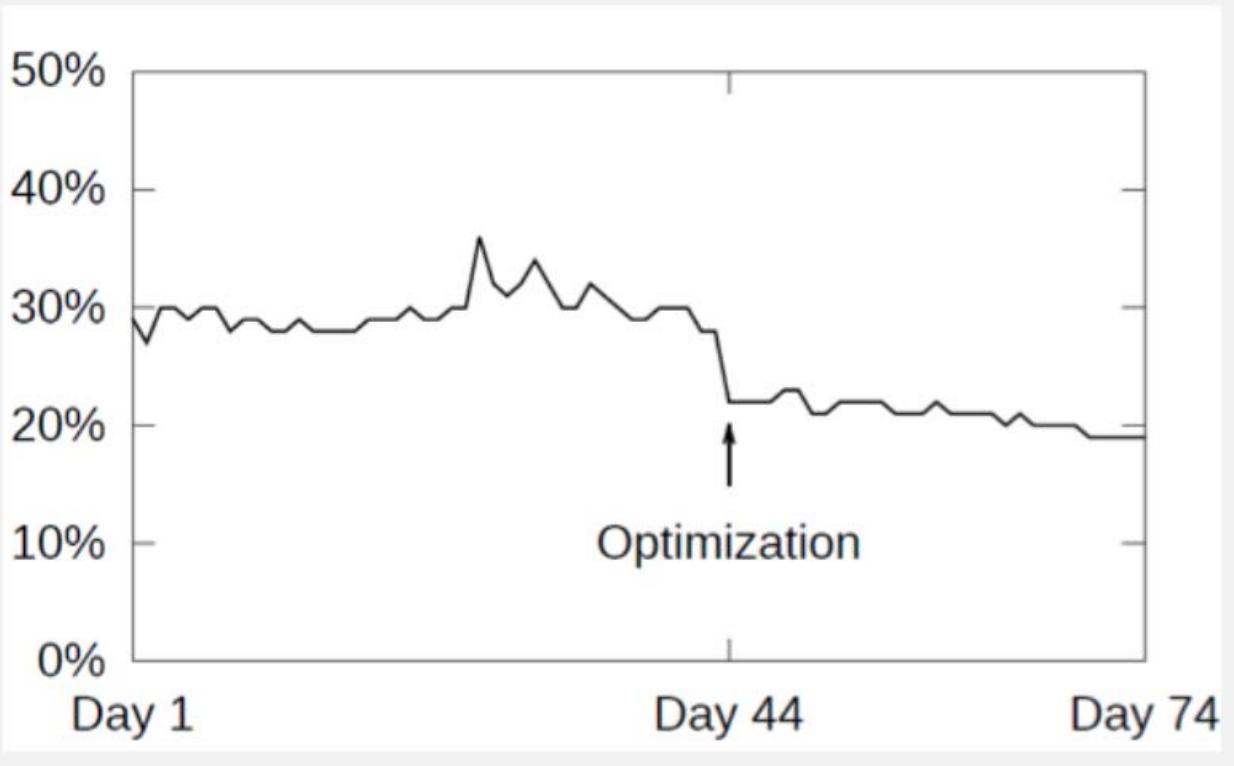
自动挖掘模块报警事件之间的关联关系，这需要对大量的事故单和项目流程进行学习，从而达到自动根因定位

场景二：自动检测PV异常

主要是学习PV的变化曲线，标注出异常点；当数据量不足时，也可以人为模拟一些日志，制造异常的峰谷值。



场景三：自动分析性能瓶颈并提出优化建议



场景四：自动关联KPI异常与版本上线

KPIs（Key Performance Indicators）是用来衡量服务性能的关键指标。服务器的异常行为包括潜在的风险、故障、bugs、攻击等。而KPI异常检测是指在KPI时序曲线上识别异常行为，并进行诊断和修复。

场景五：自动评估软件更新对应用的影响（PV、网卡吞吐率、内存利用率）

场景六：从线上事故报告单中提取有价值的信息

现在事故报告单在公司里都有相对固定的格式，可以用NLP技术+关键词提取+命名实体识别等对事故报告单结合PV、KPI等指标进行分析。

上述情景下这几大要素的解决：

- 数据：海量日志作为特征数据。
- 标注：格式化的线上事故单、事故报告。
- 算法：运维人员向算法开发人员描述运维异常，开发人员负责构建异常检测系统和检测器。
- 应用：运维人员可以设计、部署、使用、并受益于智能运维系统，形成有效闭环。

同时，如果我们把关注的那些KPI，如果抽象成时序数据，跟电商的销售数据，跟游戏的KPI指标没有本质的区别，因此，如何结合行业，也可以做一些智能的销售预测，游戏在线峰值监控等。如果抽象成算法层面，可能都有很好的应用场景，但是如果在算法层面进行更多投入，可以跳出运维本身到智能运营这块。

智能运营

微信自动化运营工具及微信读群助手。

场景一：微信读群助手：每天群过多，大多数群处于『消息免打扰』的状态，希望智能的读取群中的有用信息，生成每日群消息日报。

数据及标注：有很多hook的方式可以获取到个人的所有群的消息。对群消息进行人为的划分，实际上抽象为『有用』和『无用』的二分类问题。

例如，在『人工智能』的相关群中，对讨论AI相关的聊天，划分为『有用』，而『呵呵』或者表情，划分为『无用』。在『吉他弹唱』相关的群里，将吉他相关的聊天以及分享文章划分为『有用』，其他划分为『无用』。

模型及算法，可以参考利用深度学习进行情感分析的论文以及开源项目。例如：

https://github.com/yala/introdeeplearning

但是，聊天不同于陈述。这背后涉及到一个本质的区别——“是否有交互”。一旦涉及到交互，情感分析（sentiment analysis）评判标准的复杂度就要上升不止一个数量级了。

原因第一是大部分隐藏信息不出现在文本里，第二是交互对话信息的跳跃，这导致LSTM的记忆其实作用不是太大。

一些方式可以利用NLU语义理解，从文本中抽取重要的实体和意图，作为特征加入到学习中，并且加上一套规则，结合用户的建模模型再对这个聊天交互的『作用』进行判别。

场景二：微信自动化运营工具：如果你是一个B端，如何同时管理多个用户群。除了目前的第三方工具的微信群的消息转发功能，其实有很多地方都可以引入智能管理。

例如，群内用户发广告问题，可以抽象为二分类问题，利用NLP等技术对广告进行识别。

再如，我们可以抓取分析群内用户的历史消息行为，分析他们的职业、年龄等信息，使用深度学习对用户画像标签进行建模。

参考文献：

http://www.aclweb.org/anthology/W15-1705

这样，在用户数极大的情况下，希望能筛选出最可能消费产品的潜在用户。

智能测试

智能测试分为两种，一种是采用智能化的手段对线上产品做测试，另一种就是本身对深度学习模型引入深度学习的方法进行测试。

场景一：比如新上线一个功能，从UI界面的操作角度，可以有自动化的测试流程。顺次执行界面上的各个功能，统计是否达到预期。再比如对各种API接口的测试，对传入接口

的参数依次做校验，以及对结果数据是否达到预期进行测试。目前在这个领域，还在观察智能化引入的地方，也和大家多多交流。

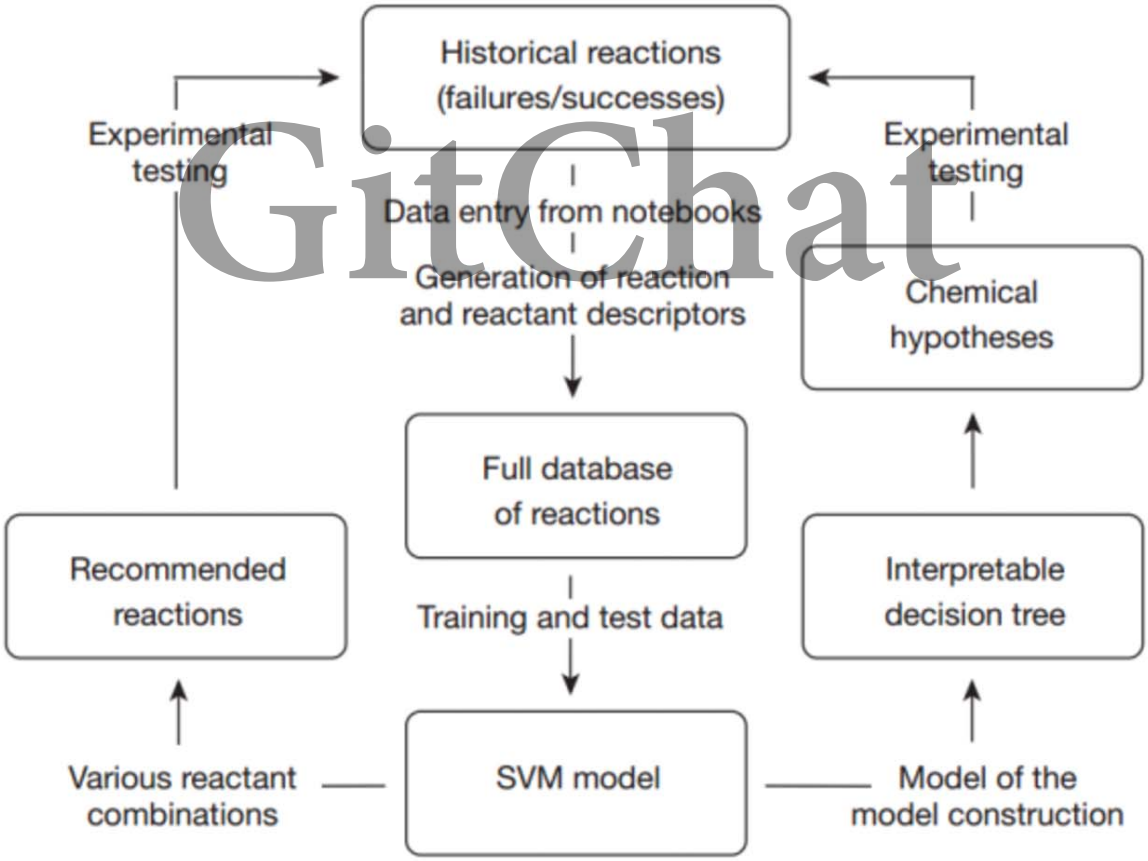
场景二：用来测试深度学习系统的工具，例如DeepXplore，一个测试深度学习系统的白箱框架。它的主要作用在于:（1）可以生成输入，生成尽可能不同的异常case，测试深度学习系统逻辑的不同部分；（2）不去人工干预的情况下，识别深度学习系统的不正确行为。并且利用多个有类似功能的深度学习系统作为交叉引证，因此避免了对错误行为的手动检查。

参考文献：

https://arxiv.org/abs/1705.06640

其他领域例如化学、制药工程与深度学习相结合

这种结合可以发生在从宏观到微观的多个层面：



例如上面这幅图，是使用SVM和决策树来发现无机-有机杂化材料，并且登上了Nature的封面。

在宏观上，例如在新药合成上，通常通过对药物分子化合物库的学习来找到有意义的药物分子结构。各种机器学习方法可以被用来预测化合物的毒理性，如急性毒性、皮肤敏感性以及各种慢性毒性，如致癌性、致畸性、对各种脏器的毒性等等。

到微观层面，就和理论化学相关。比如将药物分子抽象为图，它的原子是节点，键是边，利用分子的对称性来预测分子的性质。

文献参考:《Neural Message Passing for Quantum Chemistry》

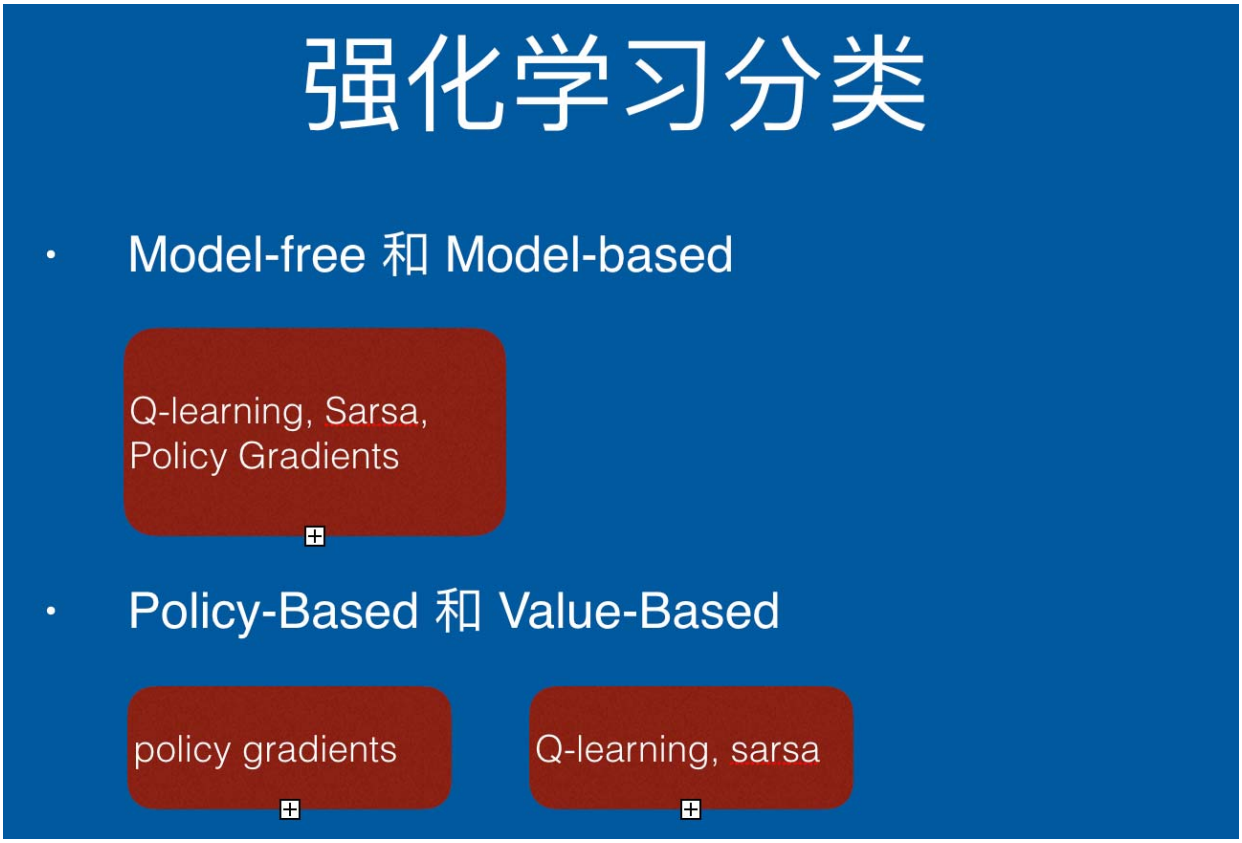
将深度学习的方法引入化学，实际上是一种统计代替解析的思想。例如，在自然语言处理领域，最初的方法是像编程语言一样，写语法规则做语法分析然后得到语法树再做下一步考虑。后来采用基于统计的方法——不考虑一个词A到底是什么成分，只看这个词A出现在另一个词B后面的概率有多大。化学也是类似，不再去使用DFT求薛定谔方程的近似解，而是基于累积的现象，用统计的方法，来预测分子的性质。

几大要素的解决：

数据及标注：即表示化合物的方法。例如毒理性，遇上一条记录一条，量不是很大。描述化合物的方法，有很多现成的软件，比如PaDEL，这些描述符会计算化合物的一些结构信息，比如包含几个芳香环，有几个sp2杂化的碳原子之类的，也会计算化合物的电荷分布还有极性等性质。

搜索和推荐

目前，推荐和搜索中引入强化学习是非常有效的手段。强化学习，目前分类可以有以下几种：
有根据奖赏最大化的不同，是基于概率（Policy_based），直接输出下一步要采取的各种动作的概率, 然后根据概率采取行动；还是基于价值（Value_based）的方法输出则是所有动作的价值, 我们会根据最高价值来选着动作。



有根据状态更新策略来划分的，比如是一个回合再进行更新，还是每进行一个动作就去更新。

主要建模方法为，把搜索/推荐系统看作智能体（Agent）、把用户看做环境（Environment），则商品的推荐问题可以被视为典型的顺序决策问题。Agent每一次排序策略的选择可以看成一次试错（Trial-and-Error），把用户的反馈，点击成交等作为从环境获得的奖赏。在这种反复不断地试错过程中，Agent将逐步学习到最优的排序策略，最大化累计奖赏。

例如在推荐商品的场景中，利用用户前几个状态下的点击、下单、收藏等行为，来推荐出当前状态下的商品列表。

参考文献:

- 《Reinforcement Learning Architecture for Web Recommendations》
- 《结合TensorFlow进行强化学习的代码实现》

视觉与行业结合

视觉和电商行业结合，已经有一些落地的产品应用。例如，陌上花科技对视频和直播平台，做实时的贴图广告、互动；玛隆科技做时尚的智能搜索，可以上传服装图片，找到含有相似服装的图片。Amazon Go也是利用视觉技术，铺设线下的无人超市。视觉和行业结合的点也非常多。

那么在AI+的过程中，我们会面临哪些问题，以及如何解决呢？

二、深度学习面临的4个挑战及递进解决方案

标注数据量较小

目前标注数据是非常昂贵的，尤其当数据量很大的时候。因此，如何从无标注数据或者尽可能需要少的标注数据里学习，一个途径就是利用生成式对抗网络，以及对偶学习的思路。

很多AI领域的任务，例如机器翻译（中英对译）、语音识别和语音合成，图像描述和图像生成，问题回答和问题生成等，都是对称的任务。

而对偶学习的思路在于，用两个神经网络分别对对称任务进行学习，用学习的结果和源数据的相似性大小来进行训练。

另一个途径就是做一些自动标注工具。用一些标注数据先训练一个自动标注模型，尽管准确度可以不是很高。用这个自动标注工具来对剩余的大批量数据进行标注，最后加入人工审核校对的过程。

模型本身太大，如何应用在移动端以及尽量不损失精度

目前在手持设备上采用AI模型是前沿趋势。这就衍生出了很多加速计算的方向，其中重要的两个方向是对内存空间和速度的优化。采用的方式一是精简模型，既可以节省内存空间，也可以加快计算速度；二是加快框架的执行速度，影响框架执行速度主要有两方面的因素，即模型的复杂度和每一步的计算速度。

精简模型主要是使用更低的权重精度，如量化（quantization）或权重剪枝（weight pruning）。剪枝是指剪小权重的连接，把所有权值连接低于一个阈值的连接从网络里移除。

而加速框架的执行速度一般不会影响模型的参数，是试图优化矩阵之间的通用乘法（GEMM）运算，因此会同时影响卷积层（卷积层的计算是先对数据进行im2col运算，再进行GEMM运算）和全连接层。

量化（quantitative），这里不是指金融上的量化交易，而是指离散化。量化是一个总括术语，是用比32位浮点数更少的空间来存储和运行模型，并且TensorFlow量化的实现屏蔽了存储和运行细节。

神经网络训练时要求速度和准确率，训练通常在GPU上进行，所以使用浮点数影响不大。但是在预测阶段，使用浮点数会影响速度。量化可以在加快速度的同时，保持较高的精度。

量化网络的动机主要有两个。最初的动机是减小模型文件的大小。模型文件往往占据很大的磁盘空间，例如，ImageNet上训练出的几个模型每个都接近200 MB，模型中存储的是分布在大量层中的权值。在存储模型的时候用8位整数，模型大小可以缩小为原来32位的25%左右。在加载模型后运算时转换回32位浮点数，这样已有的浮点计算代码无需改动即可正常运行。

量化的另一个动机是降低预测过程需要的计算资源。这在嵌入式和移动端非常有意义，能够更快地运行模型，功耗更低。从体系架构的角度来说，8位的访问次数要比32位多，在读取8位整数时只需要32位浮点数的1/4的内存带宽，例如，在32位内存带宽的情况下，8位整数可以一次访问4个，32位浮点数只能1次访问1个。而且使用SIMD指令（19.2节会加速介绍该指令集），可以在一个时钟周期里实现更多的计算。另一方面，8位对嵌入式设备的利用更充分，因为很多嵌入式芯片都是8位、16位的，如单片机、数字信号处理器（DSP芯片），8位可以充分利用这些。

此外，神经网络对于噪声的健壮性很强，因为量化会带来精度损失（这种损失可以认为是一种噪声），并不会危害到整体结果的准确度。

那能否用低精度格式来直接训练呢？答案是，大多数情况下是不能的。因为在训练时，尽管前向传播能够顺利进行，但往往反向传播中需要计算梯度。例如，梯度是0.2，使用浮点数可以很好地表示，而整数就不能很好地表示，这会导致梯度消失。因此需要使用高于8位的值来计算梯度。因此，正如在本节一开始介绍的那样，在移动端训练模型的思路往往是，在PC上正常训练好浮点数模型，然后直接将模型转换成8位，移动端是使用8位的模型来执行预测的过程。

如何从小样本中有效学习

深度学习往往需要大量数据，当数据量不足，或者数据不足以覆盖所有场景时，往往就要把深度学习、知识图谱、逻辑推理、符号学习等结合起来，将人类已有的一些先验知识结合神经网络进行训练。

例如，《Label-Free Supervision of Neural Networks with Physics and Domain Knowledge》中介绍的『用物理和特定领域知识让神经网络进行不带标签的监督学习』，在视频中把运动的枕头的轨迹检测出来。结合物质知识，枕头运动的轨迹应该是二次型的抛物线，这样就减少林要大量地对视频的每一帧枕头运动轨迹进行标注。

数据稀疏

数据稀疏性很多场景下面临的调整，尤其是个性化推荐系统中，待处理的推荐系统规模越来越大，用户和商品数目动辄百千万计，两个用户之间选择的重叠非常少，用户对商品的消费、点击、评论行为更是稀少。数据非常稀疏，使得绝大部分基于关联分析的算法（譬如协同过滤）效果都不好。

因此，一般会用一些特征提取、或者对用户和商品进行聚类的方法。亚马逊的DSSTNE（https://github.com/amznlabs/amazon-dsstne）是专门针对稀疏场景下的开源深