# 简报

- 介绍任务背景
- 工作进展
- 敬请指导

# 问题背景

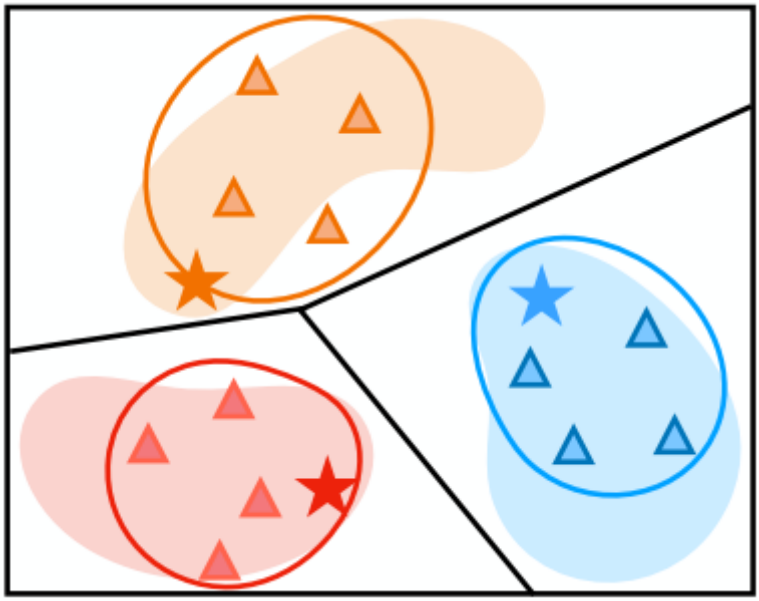
**考虑AI模型部署后的质量保证（quality assurance）问题。**相比于模型开发的场景下，解决智能软件部署后性能保证的问题环境中存在更加严格的限制。为此开发一个解决部署后AI软件质量保证工具所需要考虑的问题相比单纯设计开发设计模型更加复杂，也更加贴近实际应用。从我们工作的视角出发，主要考虑的问题有以下几点：

- **Model-irrelevant**：在开发流程中，负责AI软件基础设施（Infrastructure）的开发者与模型开发者可能并不互通。事实上，在一个软件的合作开发中，模型通常由专门的模型工程师/数据工程师开发，软件基础设施开发者负责开发能够调用模型的载体。这两者有着明确的分工，自身的专长也各有不同。因此使用一个有效的工具（比如一个Wrapper）去解决一个交付后/部署后AI模型的质量保证问题，应该**对模型本身**而言（比如模型结构等）是无关的。这样可以保证没有相关专业知识的开发者也能使用。同时也避免了模型本身产生更改，导致潜在的问题
- **Limited-access**：在实际开发流程中，软件开发者通常收到的是一个打包好的软件，Github下载的他人训练好的模型文件, 或者更甚，可能仅有一个API调用接口（考虑使用第三方提供的服务，而第三方显然不愿把自己的训练数据等信息公布）。在**无法获得充足所需信息**条件下，如何构造一个能够满足目标需求的工具成了一个非常困难的问题。在我们工作的设定中，条件更加严苛：唯一可知的**只有分类输出的置信度confidence**。这个条件设定是非常符合实际需求，也因为要求极少，使得整个工作更具挑战性
- **Operating locally**：虽然OTA等在线更新技术在各种软件中广泛应用，但也需要考虑**不可联网**的软件质量保障。为了提高通用性，开发一个能够对模型进行修复的工具时，我们更加希望这个工具能够本地化工作。解耦工具对于网络连接的依赖，使得一个部署后的智能软件在任何环境下都能实现自我更新保持性能，是工具一个理想的属性
- **Generalization**：作为一个工具/解决方案，应该能够处理**所有/或特定某类常见情况**下的问题，才具有更好的价值。因此从这个角度来看，这个工具应该是不具体到某个细分问题上。对于所有基于深度学习的分类器，工具应该在绝大多数情况下能够提供**一定的效果** (当然越高越好)
- **Efficiency** : 作为一个bridging the gap的工具, 理想情况下它应该是能够在**占用尽可能少的资源**的情况下, 提供好的工作性能. 这其中就需要在设计上考虑到资源需求和性能的权衡. 但同时面向未来来看, 随着设备算力的快速提升, 也许不用格外严格地限制整个方法的资源消耗 (?)

# 工作进展

- 把问题视为在**训练数据不可知/模型内部不可知**情况下, 利用在operational environment中收集到的**无标签数据**对整个模型进行更新的问题. 这项工作主要参照了半监督学习Semi-Supervised Leanrning (SSL), 小样本学习Few-Shot Learning (FSL), 增量学习Incremental Learning (IL) 和域自适应Domain Adaption (DA) 的一些思想. 目前SSL/FSL/IL/DA这几类里的工作都比较属于理想的setting, 主要体现在它们的setting中可以同时获得原来模型的训练样本, 即source domain数据和另一个场景中的数据, 即target domain. 这种setting让问题简化了很多, 并且其中有很多工作都是需要修改模型结构或者训练过程(比如adversarial domain adaption)
- 主线想法

1. 通过模型对其进行逆向 ( 个人叫做回溯/Retrospection) , 首先获得模型训练集中每一类数据分布的中心位置. 直观来说, 对于一个训练好的分类模型, 在模型内部即形成了每一类的不同分布区域, 如图所示



在每一类数据的边缘, 存在决策边界. 直观来说, 靠近决策边缘的数据, 其置信度会随着与到分布中心距离的增大而降低. 因此利用这个点, 使用模型进行回溯, 对每一类构造出高置信度的数据作为参考点 (称之为Retro-Anchor), 取得每一类的Anchor, 组合起来就形成了一个数据集Retro-Support Set

2. 使用数据增强方法, 如AutoAugment等对在operational environment中采集到的数据进行扩充, 其中首先对其进行weak augmentation (比如翻转等), 然后计算模型在这些数据上的预测均值, 然后在对这些数据做strong augmentation, 并且也使用在weak augmentation上输出的预测均值作为标签. 这样的原因在与strong augmentation有可能过于激进, 产生与未增强的数据不同的类别

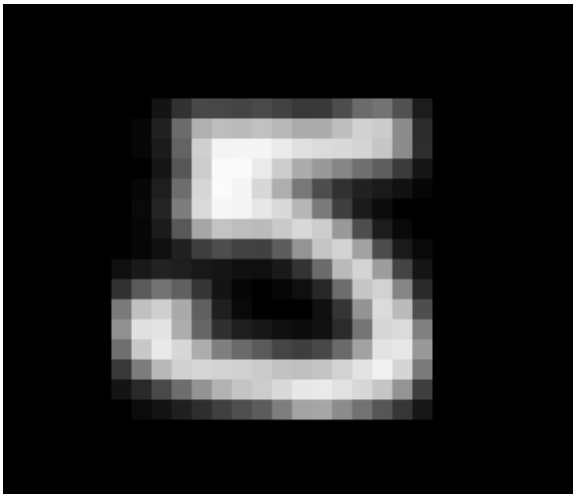
   由于Anchor只取生成数据中每类confidence最高的一个(或几个), 因此也需要进行数量上的扩充. 通过计算Anchor的统计量, 比如mean和variance. 同类的数据在统计量上应该是相似的, 且生成的Retro-Anchor靠近每一类分布的中心, 因此可以将这些Anchor数据的统计量作为生成source domain中数据的指导

3. 使用一个简单的神经网络 (或其他任何回归方法) 对Rough Set中的数据按照目标为Retro Support Set中的Anchor (包括增广数据) 进行Distribution Alignment. 这样就在两个不同domain (或者相同domain, 但未在训练集出现过的数据) 之间训练出了一个transformer. 这样即得出了两个domain之间的correspondence. 通过将采集的数据进行相应的transformation, 这样在不修改模型结构, 也无需获取原有训练数据权限的情况下, 实现对新domain的数据进行处理

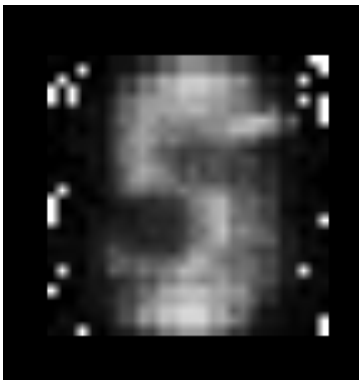
4. 由于在operational environment的数据注定是和原来训练模型的数据在分布上有差别, 并且在整个distribution alignment的过程中, 初始阶段数据量缺少, 导致并不能准确的取到数据的分布. 因此考虑可以将采集到的数据的分布中心变成moving average, 随着更多的数据被采集到, 对operational environment中数据的分布的估计也更加准确

- 一些实验图. 使用MNIST - USPS两个数据集, 将模型使用MNIST数据集进行训练, 然后使用模型预测USPS数据集. 使用USPS的某个数据 + 随机的Noise, 获得的Retro Anchor如下
  - MNIST数据集的5 (仅示例, 下同)

- USPS数据的5



- 使用USPS数据集进行回溯生成在MNIST的分类器上99% confidence的5



## 遇到的难点

- 在进行回溯训练中, 会出现模式崩塌 (mode collapse), 即当回溯的一个网络发现生成某个单一的类型的结果 (参照GAN, 因为发现某一个特定生成模式即可骗过discriminator, 因此就只生成这类模式)

  考虑在loss项加上一些regularization解决 / 或者对于每一类都单独使用一个generator (带来的开销并不会很大, 因为只需要回归生成一个99%的synthetic data作为锚定数据即可)
- 数据增强方式的选择, 对于图像数据有一些增强方法, 对于语音数据可能又有另外一些, 因此这个也是task-specific的, 理想情况下应该是要做成通用的才好. 也不太可能放弃数据增强, 毕竟生成的anchor和采集到的数据集量可能是比较少的情况
- 使用distribution alignment, 还是将采集的数据再训练? 一方面, distribution alignment方法一个明显的好处就是不需要再继续训练新的模型, 但从统计意义上计算两个分布密度的差异系数, 然后对每个数据进行scaling, 可能具体到数据个体层面性能并不好. 另一方面, 通过生成数据 (包括source domain和operational domain的增强数据)再训练模型是一个增加开销, 但更为稳妥的方法

# 个人小结

- 第一学期很快就过去了. 非常感谢王老师您对我的指导, 极大地拓宽了我的视野, 让我快速进步
- 论文方面, 考虑到我现在准备在做的工作, 主要看的是半监督学习等一些能够利用上无标签数据进行模型更新的内容, 个人觉得还需要更多地读一些从其他视角来解决此类问题. 另外软件工程方面的论文也需要多读一些
- 心态方面, 总觉得上学期有点急躁, 应该安下心来多学习一下, 好好思考
- 本学期初步打算是, 除去学校课程任务以外, 做出第一个工作的实验, 写出一篇初稿, 当然越多越好, 不能像上个学期一样很急的感觉 :)