第4周汇报

——张溢炉

1. 《Deep Reinforcement Learning for NLP》总结
2. 内容摘要

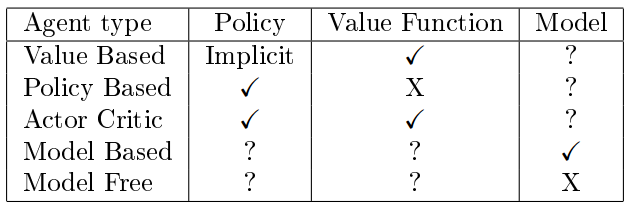
本文是对Association for Computational Linguistics一次会议的简单介绍，介绍了DRL在NLP上的应用。

1. 总结及理解
2. 最近DRL在NLP的生成、对话和信息抽取方面用的比较多
3. DRL（Deep Reinforcement Learning）在NLP应用中最大的挑战是模型的设计。
4. DRL中几个关键问题：动作空间，状态空间和奖励功能的高效实用设计；探索和利用之间的权衡；将语言结构纳入DRL
5. 启发

本文系统、全面地介绍了DRL在NLP上应用的相关作者和文章，具有很好的指导作用，可以根据自己需要展开阅读。

1. 《Deep Reinforcement Learning：A brief survey》总结
2. 内容摘要

本文首先简单介绍了DRL。然后介绍了DRL的一些方法及优化策略。

1. 总结及理解
2. 主要算法：deep Q-network (DQN)（学习玩游戏）, trust region policy optimization (TRPO), and asynchronous advantage actor critic。
3. 以前RL的缺陷：缺乏可拓展性和只能在低维应用。这是由存储复杂性，计算复杂性，样本复杂性造成的。具有强大函数近似和表示学习功能的DL出现，给这些问题提供了新的工具。DL的函数近似功能可以用于建立环境的仿真模型，可以避免机器人在真实环境中训练，大大减小的机械磨损。DL的表示学习功能可以将高维降成低维，如可以将高维的图像通过卷积神经网络降为低维，使机器人通过摄像头感知真实环境成为了可能。DRL在一些游戏中的表现已经超过人类，如AlphaGo。
4. RL可以分为如下表的几类，其中，Model-based和model-free是指agent是否有模型，模型是环境的动态和奖励的数学描述。partially observable和fully observable的区别：前者是指agent的observation只是部分状态，后者是指agent的observation包括所有的状态，其中observation是指agent通过探索观察到的状态
5. RL通过与环境交互，像人一样在试错中成长，更新自己的对环境的认知，通过奖励函数来调整自己的动作，然后形成动作集，一系列动作组成策略。当关于环境的观察满足马尔可夫性质（转换矩阵分布只与当前状态和当前的动作有关）时，RL问题被表述为MDP,其中5个重要因素是{动作集A，状态集S，状态转换分布P，贴现因子，奖励函数R}。常用回报公式如下，其中r为每步的回报，为贴现因子，取（0,1]，随着时间流逝，回报的系数会越来越小，这使得决策过程中会尽量用最少时间来完成任务。因为当其等于1时且状态无限多时，会出现回报的计算值超出计算机的计数范围，所以一般小于1。

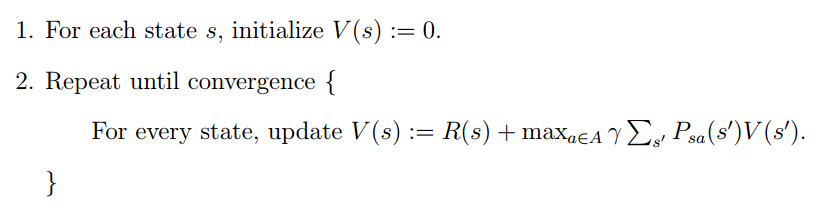
与环境交互过程中，RL会通过奖励函数不断优化状态转换分布P，从而获得最优策略，使期望回报最大化。可见，奖励函数是RL的关键。

1. MDP中的基本思想是基于下一个状态的回报作为当前状态决策的依据，这需要遍历整个环境才能知道所有状态，似乎有些不切实际。而partially observable MDPs(POMDPs)可以解决这个问题，它状态转换分布依赖于当前状态和之前的动作，其中用到了RNN。
2. RL面临的挑战：

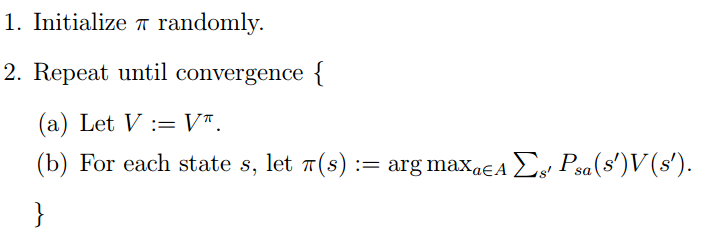
* 最优的策略必须通过与环境的交互、试错之后获得，agent收到的信息只有回报。
* agent的观察依赖于它的动作，采取不同的动作会看到不同的环境，具有强时间相关性。
* agent必须处理长时间依赖，当前的动作造成的后果经过一系列状态转换之后才显现出来。

1. 解决有限状态MDP问题的方法Dynamic programming：一般，策略迭代比较适合低维情况，但现实生活中，很多环境都是高维的，因而值迭代用的比较多。

* 值迭代：迭代步骤和优化各状态动作概率公式（Q\_learning）如下，其中Y为当前状态下的期望回报，是当前状态动作对应策略的概率分布。从式中可以看出，某个状态的回报会影响在它之前两个状态的状态转换分布概率，在点到点的任务中，当agent在地图中随意探索到目的地后，就可以通过回报反向传递，增大对应动作的概率。同时，贴现因子会使agent寻找最短路径。





* 策略迭代：策略迭代步骤如下，内循环重复计算当前策略的值函数，然后使用当前值函数更新策略

1. 文中还介绍了很多方法，自己在文中用红色波浪线做了标注，之后可以根据需要阅读文章给出参考文献深入了解，如：

* Deep Q-network(DQN),trust region policy ptimization(TRPO),and

asynchronous advantage actor critic

* heuristic search algorithm、Exhaustive Search
* partially observable MDPs(POMDPs)
* Bellman equation
* state-action-reward-state-action (SARSA)
* 策略迭代的的更新方法：先通过最小化TD误差方程更新状态转移分布概率，再遍历所有策略获得最优策略？
* 优化状态转移矩阵方法sampling：平均多个策略来设计期望回报Y？
* advantage function
* Policy search：gradient-based or gradient-free optimization
* deterministic approximations、stochastic approximations
* Actor-critic methods
* DRL误差的反向传播

1. 启发

本文让自己对强化学习和深度强化学习有了初步了解。RL可以使agent在未知环境通过学习完成相关任务。DL可以建立环境的仿真模型、将高维降为低维和通过学习近似值函数。在仿真模型中进行强化学习，可以避免机器人与真实环境交互而造成的机械磨损，同时也可添加一些先验知识。将高维降到低维，可以让图像特征更少，去掉冗余特征，让机器人识别真实环境成为可能。RL和监督学习不同的是，监督学习需要通过带标记的数据学习作预测，使其具有一定的泛化能力，但RL可以在未知环境中探索和利用已有知识选择最好动作、作出正确的决定。现实生活中的环境都是连续的，在对环境进行离散化处理之后，会产生无数的状态，这时采用暴力搜索的方法显然是不合理的，所以我们就需要综合agent长短时记忆，借助DRL作出正确决定，在无限状态的环境中更具挑战。总之，由DL和RL组成的DRL具有巨大的潜力，值得深挖。

1. 《DEEP REINFORCEMENT LEARNING: AN OVERVIEW》总结
2. 内容摘要

本文从背景、核心元素、重要机制、应用等等方面对深度强化（DRL）做了全面的介绍，对很多方法没有详细介绍，但给出了一些参考文献。

1. 总结及理解
2. DL可以用于逼近RL的值函数、状态转换分布、策略和环境模型，也常用SGD来优化DRL。DQN和AlphaGo很好的将off-policy, function approximation，bootstrapping结合了起来，避免了不稳定和分歧（divergence）的问题。
3. 探索和利用的矛盾：只利用目前已知的最优选择，可能学不到最优解，收敛到局部最优，而加入探索又降低了学习效率，但有机会找到最优解。在任务中会对两者进行折中，如-贪心算法：以概率进行探索，1-概率进行利用。其优点是直接了当，速度快，劣势是不一定找到最优策略。
4. on-policy 与off-policy的区别：

* on-policy：生成样本的policy（value function）跟网络更新参数时使用的policy（value function）相同。如SARAS算法
* off-policy：生成样本的policy（value function）跟网络更新参数时使用的policy（value function）不同。如Q-learning算法

1. DQN（Deep Q-Learning）是将深度学习deeplearning与强化学习reinforcement learning相结合，实现了从感知到动作的端到端的革命性算法。DQN非常擅长玩游戏。它的衍生版DOUBLE DQN在Atari games中能找到更好的策略。
2. Actor-Critic方法合并了Value-based(比如 Q\_learning) 和 Policy-based (比如 Policy Gradients) 两类强化学习算法。Actor 基于概率选行为, Critic 基于 Actor 的行为评判行为的得分, Actor 根据 Critic 的评分修改选行为的概率。其优缺点如下：

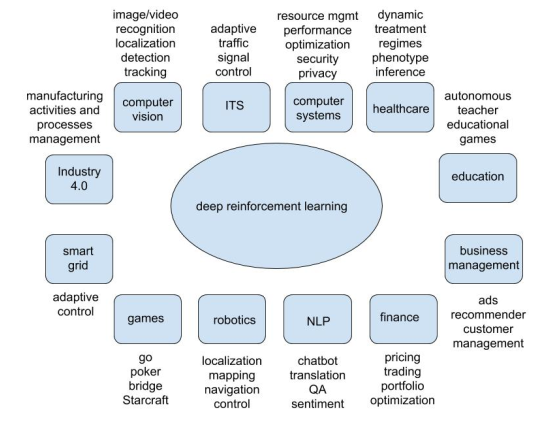
* 优点：可以进行单步更新, 相较于传统的PG回合更新要快.
* 缺点：Actor的行为取决于 Critic 的Value，但是因为 Critic本身就很难收敛和actor一起更新的话就更难收敛了。
* 为了解决收敛问题，Deepmind 提出了 Actor Critic 升级版 Deep Deterministic Policy Gradient，后者融合了 DQN 的优势, 解决了收敛难的问题

1. Deep Q-learning from Demonstrations (DQfD)结合了TD，监督学习和正则化，比DQN具有更好的初始性能，更多的平均奖励，并且学习速度更快
2. Model-free的方法需要agent自己通过采样识别环境，需要花费比较多时间，也可能物理上无法实现，Model-based的方法以数据有效的方式学习价值函数和/或策略，然而，它们可能遭受模型识别的问题，使得估计的模型可能不准确，并且性能受到估计的模型的限制。
3. 知识对于RL的进一步发展至关重要。 知识可以通过值，奖励，策略，模型，探索策略等以各种方式纳入RL。将知识加入RL，RL可以将像人一样拥有一定的常识。
4. DRL的几个重要机制

* attention and memory：注意力是一种关注突出部分的机制。 内存长时间提供数据存储，注意力是一种内存寻址方法。如DNC（Differentiable neural computers），其网站链接为：

<https://deepmind.com/blog/differentiable-neural-computers/>

* unsupervised learning：目前深度学习中的无监督学习主要分为两类，一类是确定型的自编码方法及其改进算法，其目标主要是能够从抽象后的数据中尽量无损地恢复原有数据，一类是概率型的受限波尔兹曼机及其改进算法，其目标主要是使受限玻尔兹曼机达到稳定状态时原数据出现的概率最大。DRL用到的方法有：Horde、unsupervised auxiliary learning、generative adversarial networks
* transfer learning：借助其他领域的知识来辅助学习
* multi-agent reinforcement learning：多智能体系统由一群有自主性的，可互相交互的实体组成，它们共享一个相同的环境，通过感知器感知环境并通过执行器采取行动。多智能体在现实生活中已有应用，如机器人战队，分布式控制和资源管理。
* hierarchical RL：是一种通过多层次的时空抽象来学习，规划和表示知识的方法。 分层RL是一种稀疏奖励和/或长视角问题的方法
* learning to learn：就是让机器能快速的通过学习完成新任务

1. AlphaGo的底层算法有DL、RL、MCTS(Monte Carlo tree search)
2. LEARN TO NAVIGATE可能有机会更换流行的SLAM,方法见下载文献DRL中的《LEARNING TO NAVIGATE IN COMPLEX ENVIRONMENTS》。
3. DRL的应用领域如下图所示
4. 文章最后有DRL相关参考资料，可根据需要查阅
5. 启发

本文让自己对DRL有了初步了解，它与NLP、机器人、多智能体等很多技术的结合展现出很好的发展前景。RL与DL等技术的融合能有效增强算法的学习能力，本文在differentiable neural computer (DNC)模型中使用了controller有似乎与上周MT-EQA模型中的controller存在某种联系，值得之后继续学习一下。给我的直观感觉controller像中央控制器一样，能够通过学习协调不同机制，使模型发挥更强的泛化能力，使模型具备了自我调节的能力。比如可以将controller加入GRU作为gate可能会获得更好的结果。总之，RL与其他技术的结合具备广阔的发展前景。