Relational Deep Reinforcement Learning

Vinicius Zambaldi,* David Raposo,* Adam Santoro,* Victor Bapst, Yujia Li, Igor Babuschkin,
Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan,
Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, Peter Battaglia

Contact: vzambaldi@google.com, draposo@google.com, adamsantoro@google.com

DeepMind
London, United Kingdom

【强化学习算法 33】RDRL

张楚珩
清华大学 交叉信息院博士在读

RDRL不是官方的名字，代表的是Relational Deep Reinforcement Learning。

原文传送门

Zambaldi, Vinicius, et al. "Relational Deep Reinforcement Learning." arXiv preprint arXiv:1806.01830 (2018).

特色

考虑状态空间为图像的强化学习问题（比如视频游戏），我们通常之间把图像经过CNN处理来得到低维特征。这种做法忽略了画面中不同实体相互之间的关系，因此训好这样的模型需要很多的样本，同时也只能泛化包含在训练样本分布中的情形。举例来说（不太恰当的例子），如果学会了操纵一个小兵去攻击怪物，当出现两个小兵的时候，由于智能体并不知道小兵和怪物之间的关系，因此又会把两个小兵当做另一个情形去重新学习，因此需要更多的样本；如果测试的时候出现三个小兵，智能体又会觉得这种情况没有碰到过，从而无从下手，即泛化能力不行。这篇文章就是想要教会智能体学习图像里面不同实体之间的关系，从而得到更好的sample complexity和generalization。

当然了，文章是用更为高端的语言讲的，说这是关系强化学习（relational RL），也就是要让智能体理解实体之间的关系。

文章在两类任务上面做了实验，一个是自己编的toy task叫做Box-World，另一个是SC2LE（星际2环境）里面的mini-game，都取得了不错的效果。

过程

1. 整体结构

先通过红框里面的结构把输入的一帧图像变为多个实体（entity），然后通过蓝框里面的结构学习多个实体之间的关系，最后通过绿框里面的结构把学到的东西抽象出来得到策略 π 和状态价值函数 v；使用actor-critic方法来进行强化学习，遇到要更新策略 π 或者状态价值函数 v 的时候就反向传播通过这个结构来更新里面的参数。

注意到，如果把蓝色框换成普通的神经网络block（比如residual block）这就是一个普通的强化学

习构架了。

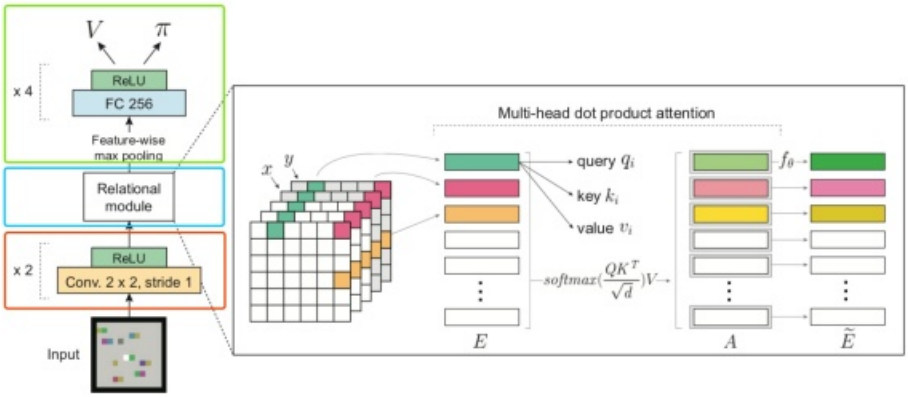


Figure 2: Box-World agent architecture and multi-head dot-product attention. E is a matrix that compiles the entities produced by the visual front-end; f_θ is a multilayer perceptron applied in parallel to each row of the output of an MHDPA step, A, and producing updated entities, $\tilde{E}$.

2. 如何定义图像中的实体？（红色框）

既然说了要学习实体之间的关系，而文章又是在视频游戏上做的实验，那么就把每个像素点当做一个实体吧。每个像素点反映的信息可能太片面，同时对于一些高清的游戏像素点可能太多。那么就把原始的一帧视频过一下CNN变成 k 个 $n\times n$ 的特征吧，这样我们就产生了 $N=n^2$ 个实体了。每个实体都表示成一个向量 $e_i\in\mathbb{R}^k$ 。

3. 如何学习实体之间的关系？（蓝色框）

实体关系的学习主要借用了multi-head dot-product attention（MHDPA，即self-attention）构架。每个实体的特征向量 e_i 都通过线性映射（权重是可学习的参数）变成三个：query $q_i\in\mathbb{R}^d$ 、key $k_i\in\mathbb{R}^d$ 、value $v_i\in\mathbb{R}^d$ ，然后通过以下方式得到self-attention的输出

$$A=\underbrace{softmax\left(\frac{QK^T}{\sqrt{d}}\right)}_{\text{attention weights}}V$$

其中 $Q,K,V\in\mathbb{R}^{N\times d}$ 是由所有实体的上述三个向量分别拼成的矩阵， QK^T 的每一行 $\in\mathbb{R}^N$ 表示每一个实体A对于任意实体的相关性，然后通过softmax选择出来相关性最高的实体B，然后通过乘上 V 就得到了实体B对应的向量，这样实体A的位置上就出现了与之相关的实体B的信息。

文章还把这样的self-attention并行做了 H 份，产生的多个 $A^h\in\mathbb{R}^{N\times d},h\in[H]$ ，然后把它们拼起来过两层MLP（图中的 f_θ ），每个实体得到和原来维度 k 一样的向量，然后再加上原来的向量 $e_i\in\mathbb{R}^k$ （residual link）。

做了上面这一大通操作，形成了一个attention block。

这样的attention block还可以叠加多层，每一层可以使用共享的参数或者不同的参数。

4. 如何形成最后的 v 和 π？（绿色框）

attention block得到的是 $\mathbb{R}^{n\times n\times h}$ 的矩阵，对这个矩阵做max pooling得到 $\mathbb{R}^h$ 的向量，然后再通过MLP即可以得到 v 和 π 了。

5. 星际环境中的特殊处理

前面讲的基本上就是Box-World环境实验中用到的结构了，对于星际环境来说，其问题结构和动作空间更复杂一些，因此需要做一些特殊处理。主要有以下几点。

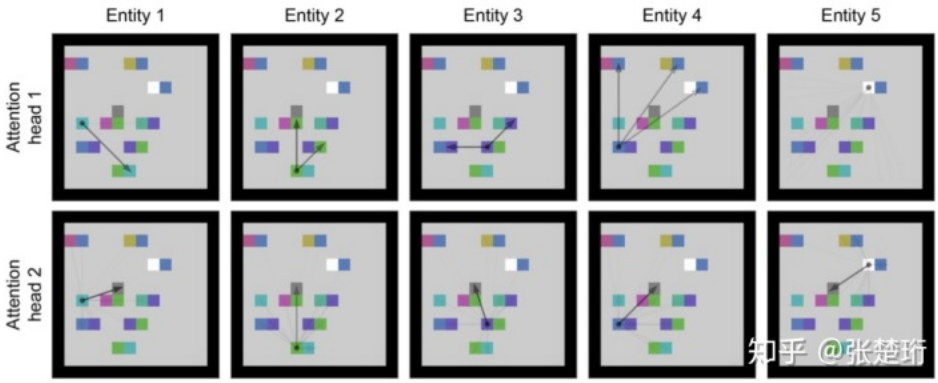
- 对于红框部分，除了使用CNN提取特征之外，还是用了LSTM让智能体能够对于最近的状态有一个认识，毕竟这个游戏从一个时间点并不能完全判断游戏的状态；
- 对于绿框部分，由于星际实验环境SC2LE的动作空间不仅包括选用一个API去调用，还需要加入一些参数，因此还需要生成这些参数。

实验

实验主要说明

- 通过在网络结构中考虑实体之间的关系，学习到的策略（渐近性能）更好；实验的对照组是把蓝框部分换成普通的residual block。
- 通过对于实体间的关系进行学习，能够有更好的泛化性能，即当画面中出现更多同样实体的时候也能够有效地处理。

这里只贴一个图，说明multi-head self-attention究竟学习到了什么。两行图分别代表两个head学到的东西，每列代表每个实体 q_i ，箭头表示该实体和其他实体之间attention的强弱，即 $q_i^T k_j$ 。可以看到，第一行里面，每个钥匙（单独的彩色块）都指向了相应的锁（两个连续块里面的右边一个，颜色和相应的钥匙相同）；第二行里面，每个钥匙都指向了小人（灰色方块）。

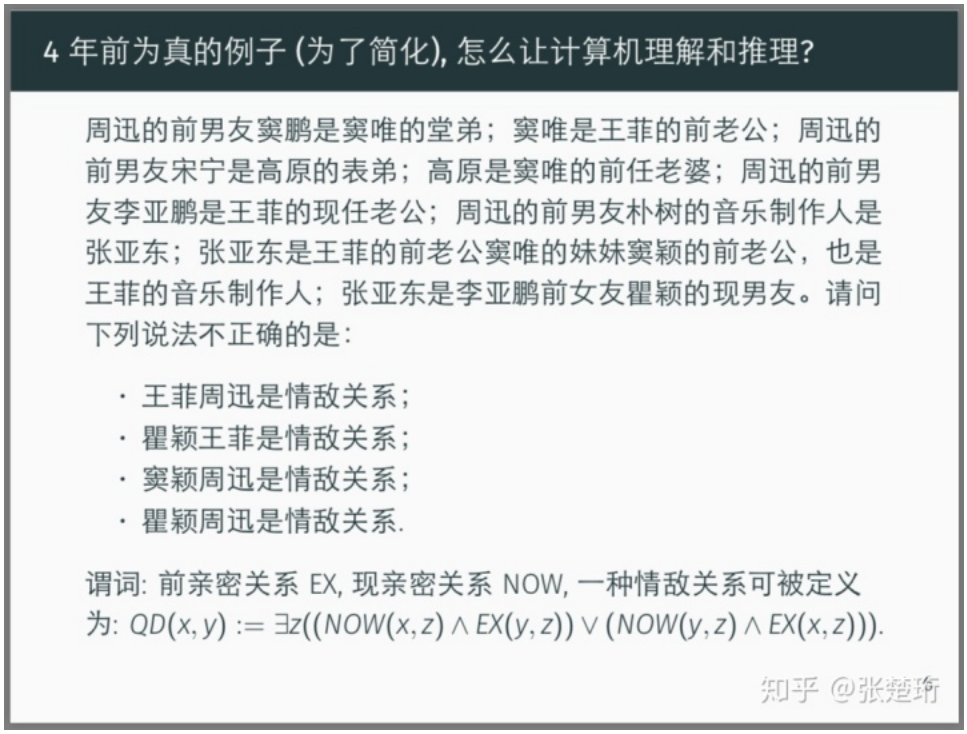


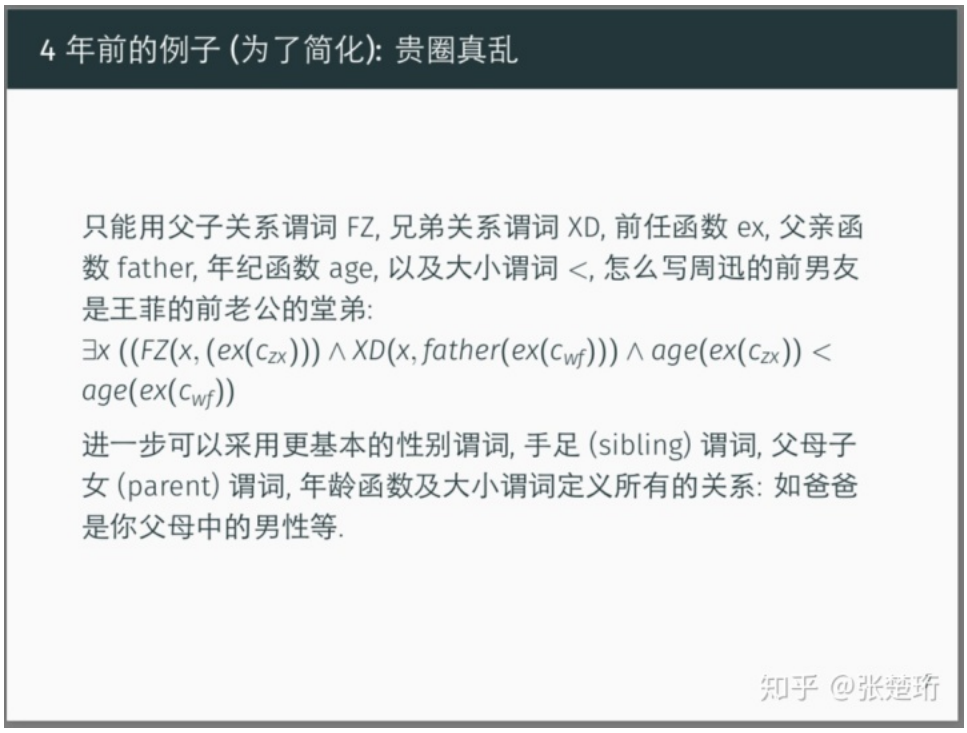
原文读不懂？

1. 关系（relational）推断什么的是很fancy的东西，我也不是很熟悉；一阶逻辑（first-order logic）什么的也很难理解；但是不懂这些不是很影响阅读。
2. 可能对于self-attention不够了解，可以参看下面这篇八个共同一作的文章 Vaswani, Ashish, et al. "Attention is all you need." Advances in Neural Information Processing Systems. 2017.（这篇文章非常值得一读，由于和本专栏主题不是特别相关，本专栏就不讲了）

彩蛋

网上搜到一个讲一阶逻辑的课件（侵删，出处phil.pku.edu.cn/persona...）





发布于 2018-11-29

强化学习 (Reinforcement Learning)

文章被以下专栏收录