王者荣耀AI即将上线

在上个月，腾讯全球合作伙伴大会上，腾讯副总裁林松涛说到，曾获得围棋世界第一的，腾讯旗下人工智能产品“绝艺”正在尝试游戏方向

未来的王者荣耀，就是你和人工智能在玩

然后当你沉浸打王者荣耀游戏时，你可能没有发觉，打败你的根本不是人类，而是人工智能

那接下来我主要讲一下在游戏AI领域的应用使用到的强化学习

强化学习在游戏AI 领域的应用

虽然强化学习在传统的机器学习分类中没有被提及，但事实上它已经能够帮助解决深度学习中很多智能控制和分析预测等领域的实际问题

比如通过用户反馈来优化人机之间的多轮对话策略

接下来我结合王者荣耀游戏AI来继续分享一些强化学习算法设计上的思路。

如何定义“类人”

AI的行为是动态、有记忆的，是能基于新的环境信息做调整的。

dota bot script 机器人脚本

如何定义“类人”

AI实现 ultimate goal (获胜) 的手段是 policy 政策，而非通过利用电脑超越人的“手速”优势来获胜

“Actions Per Minute”每分钟的按键率

王者荣耀这类游戏“复杂”在哪里

在只考虑对一个AI(agent)做强化学习的情况下（实际情况是multi-agent system, 需要协作和群体智慧），需要定义好：Environment, agent state

传统适合RL的简单游戏例如“Flappy Bird” ['flæpi]，他们的时空抽象化都是简单的，且对于goal（目标），其environment环境 是 fully observable（完全可观察的）。

2-D matrix二维矩阵

深度Q学习（Deep Q-learning，DQN）是基于增强学习中的Q学习（Q-learning）

DQN=CNN +Q-learning

CNN循环神经网络

Feature特征

DeepMind团队。低分辨率的RGB图像数据的地图和小地图

Action

不同于传统适合RL的简单游戏，如 “Flappy Bird”它的action只有上下的移动。

而类似王者荣耀这类游戏，其复杂之处在于action 多，例如基础的action：移动类的有8个移动方向，还有普通攻击类、使用技能类、买卖物品、使用物品等。

看起来基础的 action 的个数也不是很多，怎么说action多呢？

Multi-Step Plan Environment 多步规划环境

动作集是一组可能的长度动作序列

**在《Deep Reinforcement Learning in Large Discrete Action Spaces》**

**在大离散的动作空间中深度强化学习**

在这篇文章中提出用Wolpertinger 框架：先将状态与连续的动作空间对应，得到一个属于连续空间的“原型”动作(proto-action)。再把它映射到原动作空间去。

其实就是寻找k个最接近的原始动作，框架还是actor-critic的框架，多了这个离散到连续，连续到离散的转换，也就能够处理大规模的离散问题了

action embedding行动嵌入

generalize over the set of actions推广行动集

sub-linear complexity relative to the size of the action set相对于动作集大小的次线性复杂度

multi-class classification 多类分类问题

用纠错输出码把学习复杂性从

另一方面的工作是**multi-class classification** 多类分类问题

用纠错输出码把学习复杂性从

**error-correcting output codes**纠错输出码

可以参考这三篇文章中的思路方法解决

Goal

在王者荣耀里也可以参考类似做法，比如为了实现ultimate goal（终极目标）推倒对方基地获胜，

里面能举出一些 in trinsically generated goals by time scale，按时间尺度产生的内在目标

例如：5分钟左右集合团战团灭对方，10分钟左右要推倒对方几座防御塔等。

sparse feedback（稀疏的反馈） epsilon-greedy贪婪算法 intrinsically motivated 内在动机 intrinsically generated goals本质上产生的目标

learn an optimal policy to chain them together.学习一个最佳策略把它们连接在一起。

top-level module (meta-controller)顶层模块（元控制器）

lower-level module (controller)低级模块（控制器）

过程是(a) 有个top-level module (meta-controller)去选generated goal，(b) 有个lower-level module (controller) 去达成或者关闭这个generated goal，然后这个 meta-controller 反复重复(a)-(b)