理解基础:

- 深度学习基础
- 增强学习基础

创新点：构建一个全新的Agent，基于Deep Q-network,能够直接从高维的原始输入数据中通过End-to-End的增强学习训练来学习策略

成果：将算法应用到Atari 2600 游戏中，其中49个游戏水平超过人类。第一个连接了高维的感知输入到动作，能够通用地学习多种不同的task

详细分析

研究目标

General Aritficial Intelligence 通用人工智能！这绝对是人工智能当前最振奋人心的问题！创造一个单一的算法能够学习掌握执行多种任务。从而全面解放人类的重复性劳动。

Deep Q-network是本文提出的核心算法。

核心思想

使用深度卷积神经网络(deep convolutional neural network)来拟合最优的动作估值函数(optimal action-value function).

面临的困难

增强学习的困难在于在使用nonlinear function approximator非线性函数拟合时很容易不稳定unstable甚至发散diverge。
不稳定有很多原因，主要是数据的相关性太强导致小的权值更新会导致policy策略大的变化。

解决办法

- experience replay
- fixed θ 目标Q值仅周期性更新，目的是减少目标和q值的相关性。第二个办法在NIPS 2013的文章中没有

算法基本流程

Algorithm 1: deep Q-learning with experience replay.

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$

For episode = 1, M **do**

Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$

For t = 1,T **do**

With probability ε select a random action a_t

otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$

Execute action a_t in emulator and observe reward r_t and image x_{t+1}

Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D

Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D

Set $y_j = \begin{cases} r_j & \text{if episode terminates at step j+} \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{otherwise} \end{cases}$

Perform a gradient descent step on $\left(y_j - Q(\phi_j, a_j; \theta)\right)^2$ with respect to th network parameters θ

Every C steps reset $\hat{Q} = Q$

End For

End For

1）初始化replay memory D，容量是N 用于存储 训练的样本

2）初始化action-value function 的Q 卷积神经网络 ，初始的参数 随机

3）初始化 target action-value function的卷积神经网络，结构和Q的一样，参数 初始等于Q的参数

For episode = 1,M do
初始化状态系列s1，并对其进行预处理得到4 * 84 * 84的视频帧
for t=1,T do // 每个episode篇章训练一定的时间
根据概率e（很小）选择一个随机的动作
或者根据当前的状态输入到当前的网络中 （用了一次CNN）计算出每个动作的Q值，选择Q值最大的−动作（最优动作）
执行上面的动作a就可以得到reward(得分）以及下一个图像
那么下一个状态就往前移动一帧，依然是4帧的图像，再次处理得到新的网络输入
存储（上一个状态，使用的动作，得到reward，下一个状态）数据 到replay memory来做训练
接下来从D中随机选取一个存储的数据来训练网络
计算当前状态的目标action-value，根据bellman公式得到：
如果episode结束，那么就是得到的reward，如果没有结束，那么就将下一个处理好的状态输入到网络，用target网络 参数（上面的3）），得到最大的Q值，然后按下面公式计算： （用第二次CNN）

接下来就是计算当前状态和动作下的Q值，将当前处理好的状态输入到网络，选择对应的动作的Q值。（用第三次CNN）
根据loss function通过SGD来更新参数
每C次迭代后更新target action-value 网络的参数为当前的参数
end
end

具体Atari 2600 实验成果

比NIPS2013的版本改进不少，主要是Fixed target Q-network的贡献

DQN的突出表现

使用t-SNE算法来可视化高维数据，相似的state会放在接近的位置。有时候可能state不相似，但期望的reward相近。结论是

- 这个网络能够从高维的原始输入中学习支持可适应规则的表征。

> 疑问点: The representations learned by DQN are able to generalize to data generated from policies other than its own

- 能够发现相对长的策略，虽然依然无法应对很长策略的游戏

和脑科学对比

> Reward signals during perceptual learning may influence the characteristics of representations within primate visual cortex.
> The hippocampus may support the physical realization of such a process in the mammalian brain,with time compressed reactivation of recently experienced trajectories during offline periods

未来使用优先的经验进行训练必然会改进性能！

小结

Nature的文章结构和NIPS这种会议的文章结构是完全不一样的。Nature更重要的是告诉不了解的人们他取得的成果，而具体的技术实现则全部放在附录。

Method 技术方法再分析

这里只分析和NIPS2013不一样的地方

Preprocess预处理

Nature版本：使用Y通道图像(luminance亮度分量）
NIPS版本：使用灰度图像

Model Architecture 模型结构

和NIPS一样

Training 训练

比NIPS多训练了其他游戏。
使用RMSProp 更新参数（看来有必要再次学习一下Hinton的课程了）

Procedure Evaluation 过程评估

只是介绍训练完之后如何评估训练效果，比如每个游戏玩30次至多5分钟。

Algorithm 算法

这里等同于NIPS的背景介绍

总结

总的来说Nature文章对DQN进行了改进，添加了Fixed Target Q-network,提升了性能，并且对更多的游戏
了效果评估，以及进一步发现了DQN算法的优点及类人特性。