需要解决的问题：

1. 爬坡怕不上去，学不会，怎么能让他爬上去？
2. 用80\*80\*4的黑白图片进行处理是否合理？或者改变黑白图像大小或者增加黑白图像帧数作为一个state，要不要改为彩色图像处理，效果会好些
3. 卷积网络层数是不是设计的太少了，对游戏的整体把握性不够？
4. 看看整个网络的设计思路有没有问题，从头到尾debug一下
5. 能不能将整个游戏的reward用来优化？
6. Double DQN还没有用，后面加上去
7. 对图片进行零中心化处理



1,用retro获取游戏图片

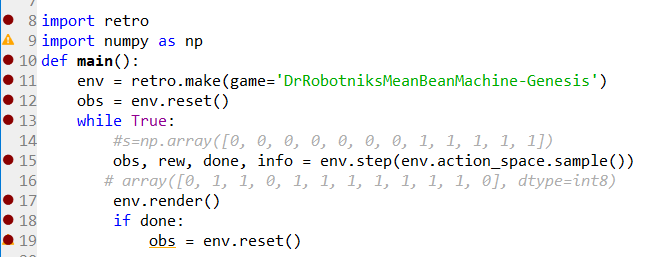


图1

x=env.render(mode='rgb\_array', close=False)得到的是在执行了某个动作之后游戏中的图片，具体可以参考env.render()函数的输入，当mode='rgb\_array'时，可以获得屏幕上的图片，然后可以用图2中的模块显示一下。



图2

2、如何解决一个游戏中包含若干个动作，而必须选择若干个动作的组合，比如向左走是一个动作，跳跃是一个动作，我们可以把向左走加跳跃看成是一个动作来执行，即组合动作算成一个单独的动作如图3

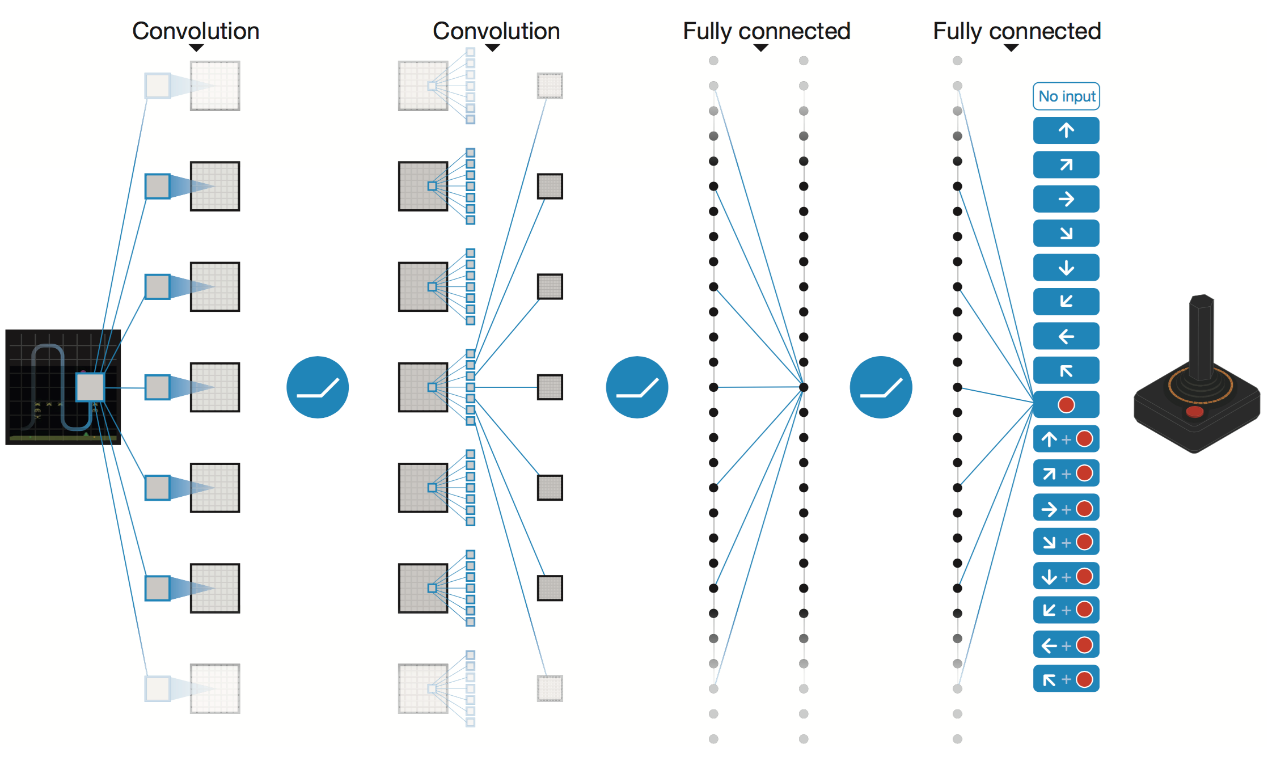


图3

3. SonicTheHedgehog-Genesis游戏的动作，共有12个二进制动作值，先分别列出如下

（1）[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 🡺静止

（2）[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 🡺静止

（3）[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 🡺静止

（4）[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0] 🡺静止

（5）[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0] 🡺静止

（6）[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0] 🡺静止

（6）[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] 🡺抱头蹲下

（7）[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]🡺左跑

（8）[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0] 🡺右跑

（9）[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0] 🡺静止

（10）[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0] 🡺静止

（11）[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] 🡺静止

（12）[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] 🡺静止

（13）

4.关于如何构造DQN，就是把输入状态变成一幅幅图片，然后进入神经网络之后输出动作，并且环境中已经给出了step函数，当前动作即可输出对应的s\_(图片)，reward, done, info.

5.关于人物在原地没办法前进的问题，把向右走设置一个reward，这个reward设置的小一点，比如0.1，只要能让他有向右走的趋势就行

6.在游戏中获得分数会转换为step中的奖励，比如游戏中获得了100分，那么step(action)中reward=100。但是回合结束没有奖励衰减，reward=0，最好是在回合结束之后有一个负的奖励值