实验报告

学号: _2017326603075_ 姓名: _陈浩骏_ 班级_17 信科 1 班_ 成绩_____

【实验目的】

解决 OpenAI Gym 中的 cartpole-v0 问题

【实验内容及要求】

采用深度学习、启发式搜索、进化算法、强化学习中的任一种或两种，解决 gym 的 cartpole-v0 问题。

【算法描述】

神经网络的搭建采用了 Keras 的 API.

首先，observation 是由 4 个值组成的 list，分别为小车位置，小车速度，棍的偏移角与棍的远端线速度. action 是一个 binary 值，他控制小车向左(0)或向右(1)移动从而将上方的棍置于平衡态. 每个 frame 结束后，若棍未达到阈值条件(倒下)则 reward+=1，solved 的定义为在连续 100 次根据 observation 给定 action 的 episode 里达到平均 reward >= 195. 其余参数与详情详见(openai-gym-cartpole-v0).

在本次实验过程中，首先拿 games_for_learn=1500 局游戏，采集 observation，与 action，这里需要注意的是，由于 observation 是传入 action 后的 step 产生的，在模型的建立过程中，上一次的 observation 与本次的 action 才有关联意义.即(last_observation 驱动 action，即传入 last_observation 以 predict action).拿 games_before_learn=1200 次随机游戏采集数据，清洗并采用 satisfied_score=57 以上的游戏策略提取特征(即高于上述分数的行为才”值得”学习).

清洗过程中，对于 action 做一次 one-hot 编码，让神经网络最后一层不采用 binary 输出，转而采用激活函数为 softmax 的二维全连接层输出后，再拿 argmax.

训练完神经网络后，根据要求，predict 100 次游戏，取平均 reward 输出.

【算法实现】

```
network = input_data(shape=[None, len(train_X[0]), 1], name='input')

network = fully_connected(network, 128, activation='relu')

network = dropout(network, 0.8)

network = fully_connected(network, 256, activation='relu')

network = dropout(network, 0.8)

network = fully_connected(network, 512, activation='relu')

network = dropout(network, 0.8)

network = fully_connected(network, 256, activation='relu')

network = dropout(network, 0.8)

network = fully_connected(network, 128, activation='relu')

network = dropout(network, 0.8)

network = fully_connected(network, 2, activation='softmax')

network = regression(network, optimizer='adam', learning_rate=learning_rate,

loss='categorical_crossentropy')

model = tflearn.DNN(network)

model.fit(train_X, train_Y, n_epoch=epoch, snapshot_step=1000, show_metric=True)
```

根据 satisfied_score 的不同， 以及每次随机游戏的质量， 输入的向量维也会发生改变， 在上述参数下， train_X 的维数大概在(2000， 4， 1)左右以供学习. 全连接层会将其映射并压缩到一个新的维度上以提取特征值， 再输出一个类似 one-hot 编码二维的 predict， 取 argmax 即可知.

【性能分析】

在压缩 episodeBeforeSolved(即本实验报告中的 games_before_learn)的时候, 发现在 1000 次到 1200 次左右都能得到接近 200 的平均 reward. 并且在调节参数时,

发现该模型有较高的损失函数值， 且该模型要注意避免过拟合， 都是由于这个模型的特殊性， 因为它的输出结果越接近游戏(即游戏完成度越好)， 它的 action 集的平均值就越趋向于 0.5, 所以导致在激活函数中总会出现微小的偏差.

在调整参数后, 可以达到相对稳定的 reward=199 的输出, 认为该问题已 solved.

【实验小结】

虽然仅取 observation 四个特征给进神经网络模型中进行训练, 感觉起来相对比较暴力， 但是该算法在数据量相对较小的时候也能得到较好的输出， 在最初建立模型时， 我曾想用大约过万次的数据来进行训练以达到结果. 但是该训练模式不足的是, 它太依赖于随机游戏时产出的数据了, 即要求越高, 数据越少, 训练效果越差, 毕竟 gym 对于该游戏的算法评分是“采用越少 episodeBeforeSolved 越好”.

运行 console 位于 console.log 中, 源码位于 gym-tf.py 中.