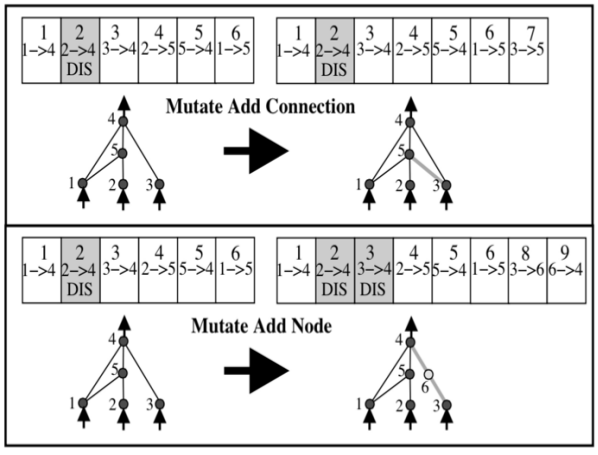
算法设计与分析Project报告--利用神经演化训练马里奥游戏机器人

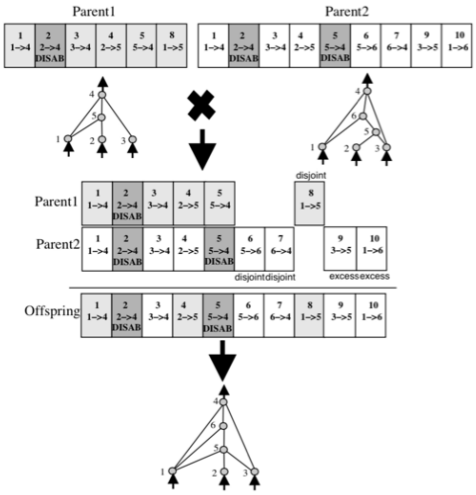
摘要

随着深度学习的日益火热，人们利用神经网络方法解决了许多现实中的人工智能问题，比如视觉，语音识别，模型预测等。卷积神经网络以训练完成后运行快速，随着数据规模增大准确性逐步上升极限很高而著名，因此在特别多的应用场景下十分好用。但是在现实生活的很多问题里面，传统的神经网络训练方法就不适用了。比如控制机器人走路，需要构建多大多复杂的神经网络才能完成这个任务？在任务过程中用什么参数作为反馈值来训练网络？这些相比于传统的训练方法来说都是很大的困难。

与此同时，有研究者开始借鉴与自然界生物进化而模拟出的遗传算法，并运用于神经网络的训练中，并把此类算法成为神经演化。这种算法只需要非常少的反馈量和很少的对该问题的预先设计就可以完成非常复杂的任务，在机器人控制，工业操控中运用广泛。本课题中我们将梳理现在有的神经演化算法，并自己实现其中的一种算法来训练完成一个能够出色地玩马里奥游戏的机器人。

神经演化算法简介





神经进化，或神经进化，是一种利用遗传算法训练的人工神经网络的机器学习的形式。所谓遗传算法就是模拟生物种群物竞天择。每一个训练单元作为一个个体，许多个体形成一个种群。每一个个体的参数各不相同，并且利用某一种编码表示为基因型。基因有变异和杂交两种方式，分别可以改变自身的表现型和将自己的表现型与其他个体结合传给下一代。通过某一种测试得到每一个个体的适应度，根据适应度高低来选择这些个体该如何变异杂交，并淘汰不适应环境的个体，

声称下一代种群。如果基因型编码和杂交的方式比较好，优秀的基因就可以一代一代的传递下去使得种群个体的适应性逐步增高直到达到期望的值。

在神经演化算法中，个体是一个个神经网络。不同的算法的区别在于他们对网络的编码方式，杂交、变异方式。最经典的神经演化算法是NEAT（NeuroEvolution of Augmented Topologies），该算法讲网络的参数以及结构都编码进基因型，变异不仅会改变网络边的参数，还能修改网络结构。也就是说最开始的时候甚至可以只加入一个点，而随着网络的演化逐步网络会变得复杂而适应当前问题。

在NEAT的基础上，研究人员们又开发了更高效的HyperNeat等算法。本课题只实现NEAT一种。

课题计划

我们的目标是训练一个能够玩马里奥的神经网络。网络将会通过一个脚本和一个实时的任天堂模拟器通信，接受当前游戏状况并实时的作出反应。为了方便网络处理，我们利用opencv对游戏图像进行处理并抽象成一个19*19的矩阵。矩阵中3代表马里奥，2代表敌人，1代表障碍物。为了能够反应游戏的实时状况，我们将上一次给出的输出，当前所在的位置以及这个矩阵作为输入，获取网络得到的输出参数。输出参数有4种，分别为前进，后退，跳跃，下蹲和发子弹。网络持续的玩游戏直到输掉。对于马里奥游戏，虽然最后的评判结果时最远距离，但是收集金币、蘑菇和杀敌也是玩好游戏必要的子任务。因此我们记录最后所到达的最远距离，收集的金币数，杀敌数，时间等来计算适应性。


```
0  0  0  0  0  0  0  0  0  0  0  0  1
0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  1  0  0  0  1  1  1
0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  3  0  0  0  2  0  0
1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1
0  0  0  0  0  0  0  0  1  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0
```

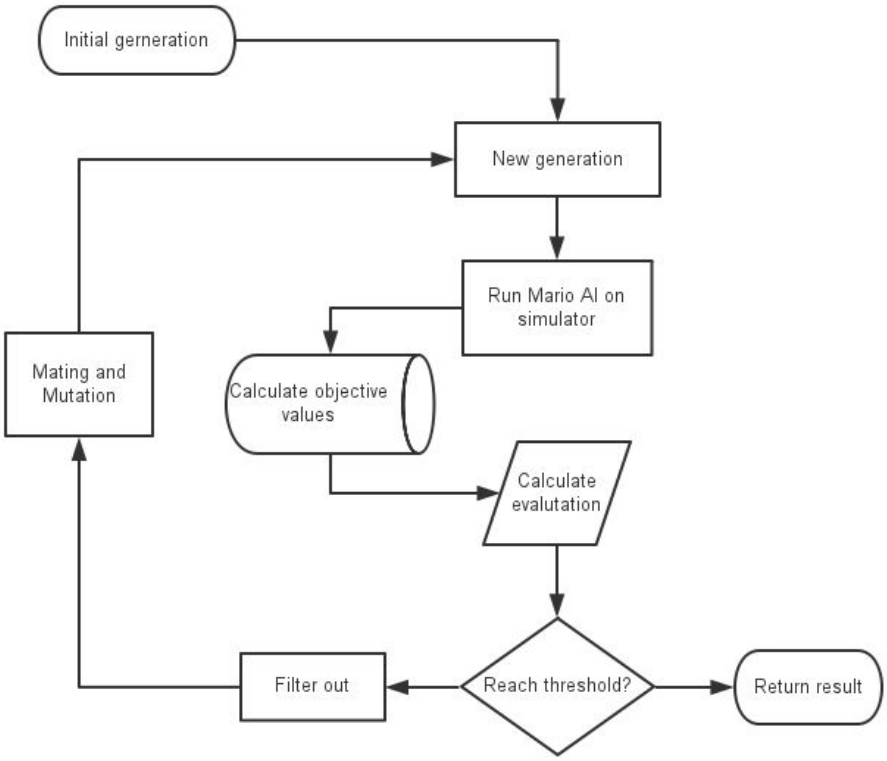
我们的工作分为三部分。第一部分是实现一个通用的遗传算法框架。这个算法框架以个体特性，问题求解适应性的方法作为基本输入，和之后的神经网络应该是松耦合的，也就是说可以利用别的问题来测试这个框架这个遗传算法框架应该做到多个个体能够并行进行测试，支持多种演化的方式。第二部分是网络的编码、变异和遗传方法。我们参考了论文中的基本思路和github上已经实现好的开源的neat库，稍作修改作为我们的网络部分。第三部分是模拟机的调试和脚本编写，并整合之前两个独立的工作设计我们的训练架构。

最终的算法流程如下：

（1）随机生成一组初始网络，作为第一代种群
（2）遗传算法框架并行地对每一个网络开启一个虚拟机去跑游戏，并最后得到脚本返回的该网络的适应性结果。
（3）根据（2）中得到的适应性结果和适者生存的远离，淘汰适应性较差的，并让适应

性好的有更高的几率繁殖保留他们的基因。同时随机的对这些个体进行变异。在这个过程中下一代种群的网络参数、结构都会发生变化。

（4）重复（2）和（3）。随着一代代的演化，适应性逐步增加，直到达到我们的预期后停止迭代。



实现要点

（1） 由于跑模拟机消耗内存资源较大，单机上最多只能同时并行跑16个虚拟机程序。对于种群个体数量较大的情况，我们采用分组训练的方法。每次训练十六个，最后再汇总

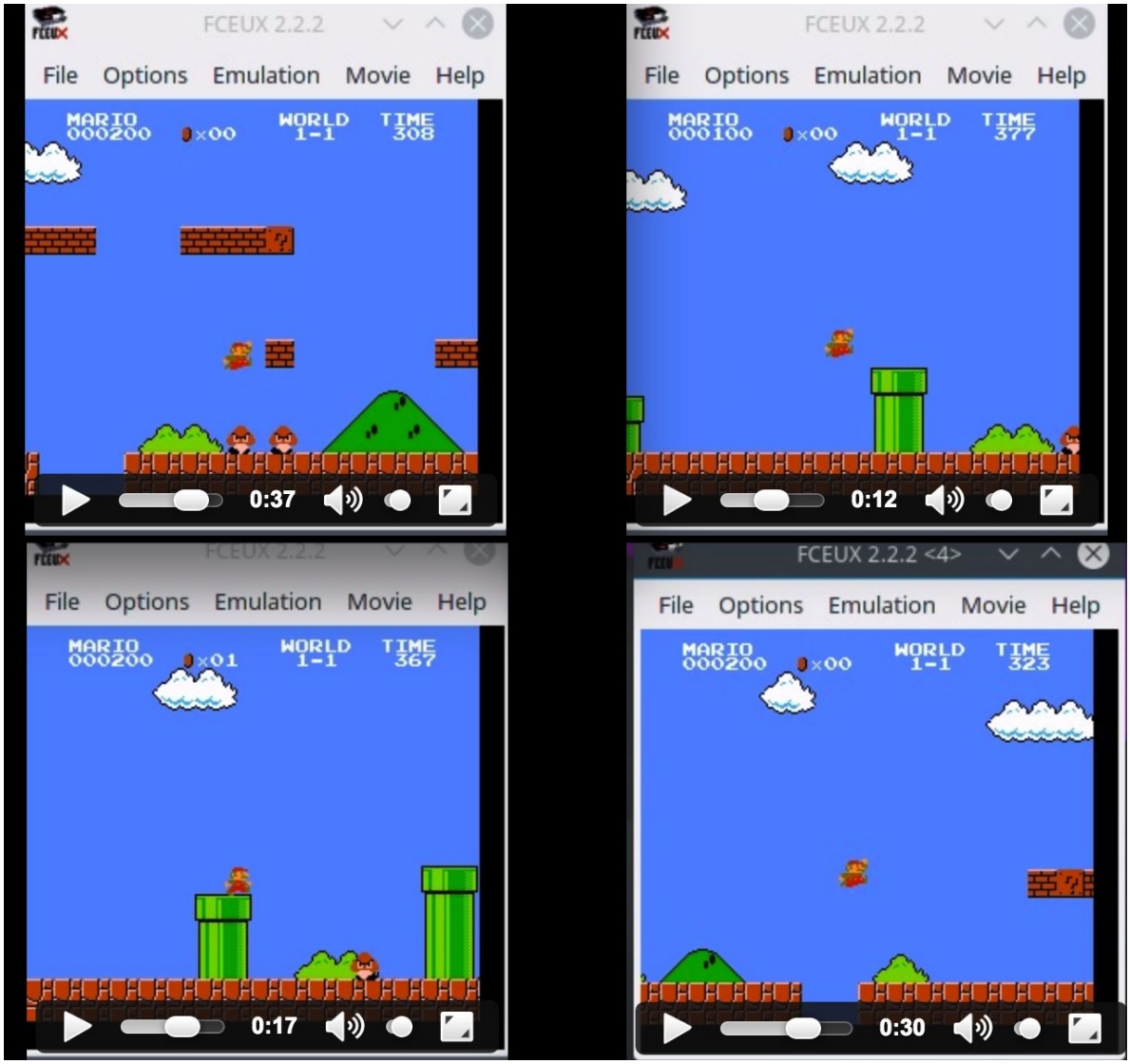
（2）遗传算法框架可以使用精英主义或者保护主义来调控淘汰模式。这些实现细节可以参见我们的源代码。

（3）在脚本编写的过程中由于和模拟器的通信是阻塞的，不能像函数调用一样得到结果，而是需要通过维护一个文件流进行通信。我们使用一个文件作为缓冲区来进行和模拟器的通信。

结果展示

（1）种群子代个数16，变异率0.3，杂交率0.3，适应性＝最后到的最远距离

我们总共对网络迭代训练了五百次。在前两百次可以看到网络非常明显的进步，到两百代时已经走到了游戏的中部。但是两百代到五百代基本没有进步。分析原因大致是两百代时网络规模已经较大，利用随即方法进行变异，出现良性基因的可能性较小。因此训练结果基本达到了饱和。

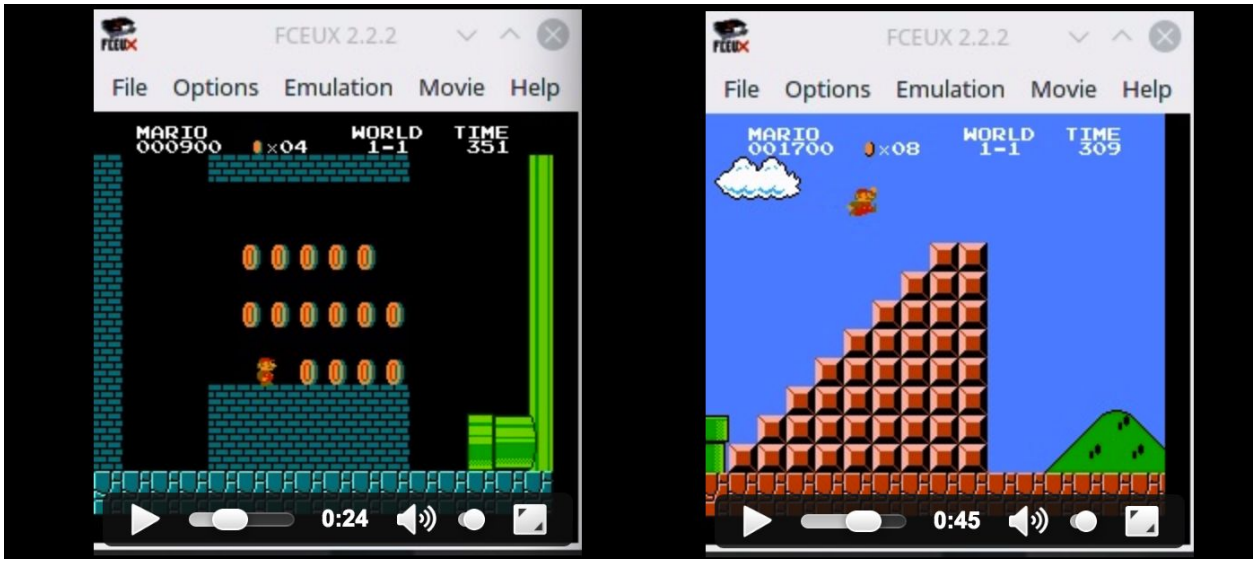


同时我们发现因为我们的评价函数只考量了最后到达的距离，在一个位置马里奥原地等待了十几秒，完全是没必要的等待。而路途中也没有去收集金币和蘑菇。因此我们修改了适应性函数的计算方法进行了第二轮训练

（2）其他参数不变。适应性＝最后的距离＊最后的距离 / 时间 ＋ 10 ＊ 杀敌数 ＋ 30 ＊ 收集蘑菇数 ＋ 收集金币数

对这种方法训练了两百代，马里奥明显学会了收集金币和渴望杀敌，同时中途也很少无谓停顿。但最后的距离没有比（1）中有更好的突破。我们认为理由还是网络庞大以后随机性太强，变异出良性基因的概率太小。我们每个种群的子代数也过小，很容易陷入过拟合的情况。

神奇的一点是用这个方法进行第二轮训练时，到了第十代的时候马里奥学会了钻地道～钻出地道后接近通关了。这也侧面体现出了遗传算法的随机性，以及保证样本多样性的好处。



分析讨论

总结来说，神经演化算法的好处在于即便对于特别复杂的问题，反馈数据很少，了解也不多的情况下，也可以提供一个通用的解决方案。并且在训练初期能较快的到达一个不错的水平，但之后的提升就越来越缓慢了。劣势在于：（1）很容易抹杀样本多样性。由于庞大网络的可能组合实在太多，如果一个种群倾向于保留某几个优势个体，很容易陷入局部最优解，而丧失了多样性。（2）随着网络的复杂，变异的效率开始逐步降低。到了两百代以后的变异，即便有可能是良性变异也不能很好的展现出结果，导致变异流失。且恰好是良性变异的概率也因为网络规模的庞大而变得很弱。

对于以上两点，第一个问题的解决方案可以是采用分割任务，多种群训练的方法。比如对于马里奥问题，为了最终达成目标，需要掌握跳跃障碍物，获得道具，杀敌等多个子技术。可以先分别对这些小的问题训练网络，再将他们放入一起杂交，并训练一段时间，再分别训练。如此反复可以保证样本的多样性。更重要的是可以完成子任务的网络结构可以在大网络中找到对应，以此对于我们理解神经网络的构造和远离会大有好处

第二个问题的解决方案是在遗传算法中加入删点的机制。现在关于神经网络节点冗余性的研究正在进行，如何评价神经网络中的某个节点是否是对正确结论至关重要的？有的人利用贝叶斯统计从最后结果反推回每个节点重要程度，或者判断节点间的相关性等方法。借用他们的方法，对于一些估计不太重要的节点尝试删去，可以有效的控制网络的规模，从而使变异能够更有效的发生。这个可以先显著改善由于网络复杂造成的训练效率低的问题。

通过这个课题，我们了解了神经网络和遗传算法的相关知识，阅读了一线研究者的相关论文。在实际操作中，自己实现了相关算法，并针对一些并发，通信的问题通过查询资料和讨论想出了解决方案。这些经历极大的提升了我们的实际编码能力和解决问题的能力。最后的几个未来展望我们认为还是很有研究价值的。如果未来有志于从事神经网络的研究可以以此作为科研目标。

参考文献

[1]Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. Evolutionary Computation, 10(2):99–127, 2002.

[2]Lars Solvoll Tønder & Ole-Petter Olsen Multi-Objective Neuroevolution in Super Mario Bros.

[3]Matthew Hausknecht, Piyush Khandelwal, Risto Miikkulainen, Peter Stone. HyperNEAT-GGP: A HyperNEAT-based Atari General Game Player