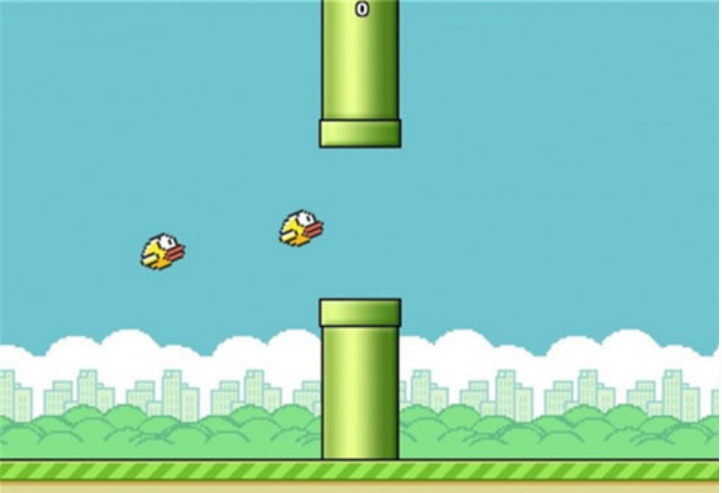
**智能计算课程论文**

**--智能计算在游戏AI领域的应用**

**一、问题提出**

目前AI在LOL、围棋等竞技游戏中表现超越人类的情况已经不在少数。然而，我们更多地只是听说这样的新闻，却对其底层的原理以及应用范围知之甚少。那么是否存在一种可能：自己动手搭建出一个能够自训练的模型，将其应用于某个简单的游戏（如Flappy Bird），最终达到一个相当高的表现水平呢？

**二、问题分析**



Flappy Bird游戏规则：

Flappy Bird是一款简单又困难的休闲游戏，玩家需要在游戏中控制一只小鸟跨越不同高度的管道组合而成的一系列障碍物。小鸟在没有操作的时候会由重力原因自由地下坠，玩家的操作就是简单地点一下屏幕或者按一下空格，小鸟便会往上飞一段距离，当点击频率超过一定值后，小鸟总体呈现往上飞得越来越高；当点击频率低于一定值后，小鸟总体呈现往上飞得越来越低。如果小鸟不慎掉到了地面或者触碰到了管道等障碍物的边界，则会被判定为游戏失败，该游戏回合结束。小鸟在一次游戏回合中前进的距离即为玩家的得分，若玩家操作精准，便能在较长一段时间内保持小鸟的飞行，从而获得较高的分数。

训练计划：

初始生成若干个模拟大脑的神经网络。在不断的迭代中给予每代遗传基因型一定的变异，每只小鸟所对应的神经网络不断进行自适应和自然选择，最终筛选出能够顺利通过尽可能多的缝隙的小鸟（优质基因型）。劣质的神经网络会很快触碰到障碍物而被淘汰掉，而优质的神经网络则会被遗传到下一代。

**三、算法选择**

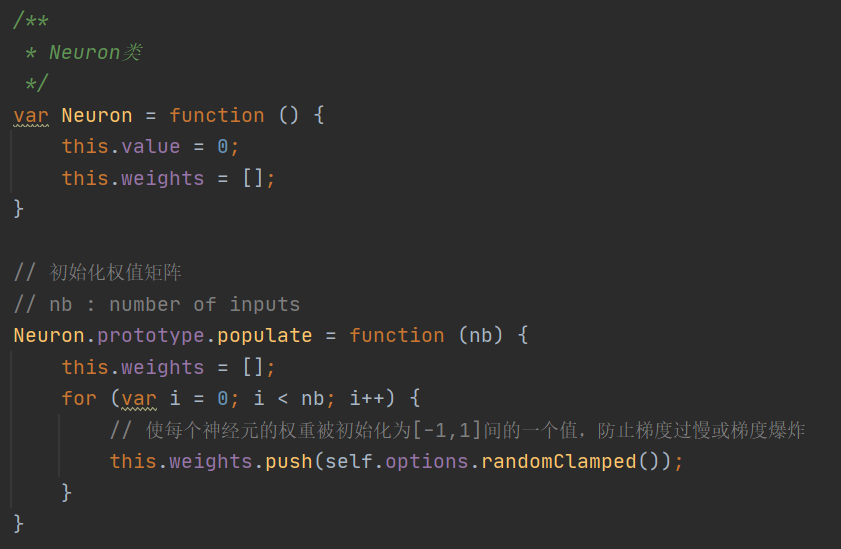
对于本问题，我使用遗传算法 + 神经网络的组合方式加以研究。使用神经网络作为每只小鸟接收输入后进行决策的方法，使用遗传算法对小鸟，也可以说是神经网络种群进行繁殖、变异和筛选，以得到较好的结果。

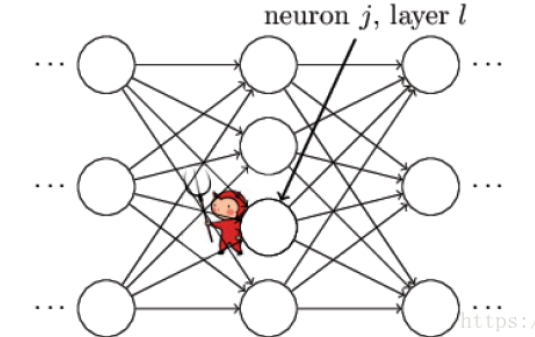
关于编程语言，由于选择使用HTML5的canvas画布来进行Flappy Bird的展示，对应编程使用JavaScript实现。

**四、算法实现 -- 构建神经网络遗传进化系统**

**1、神经网络 -- 构建每只小鸟的神经网络**

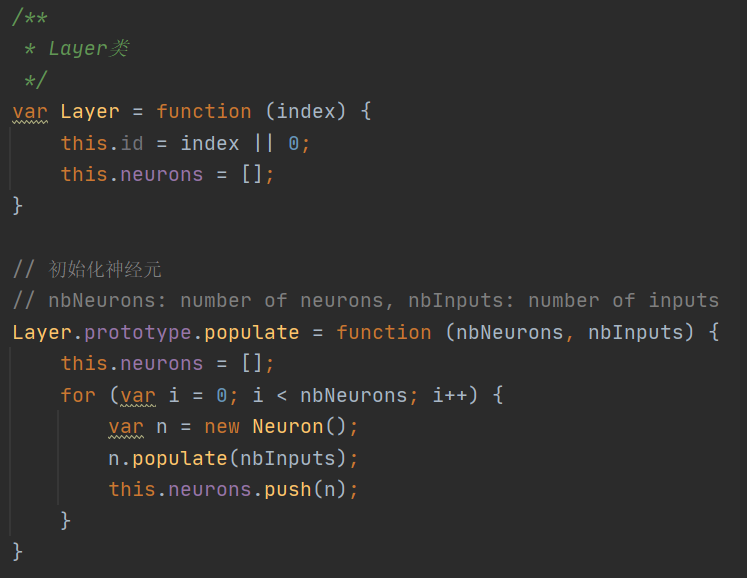
**1.1 构建神经元类**





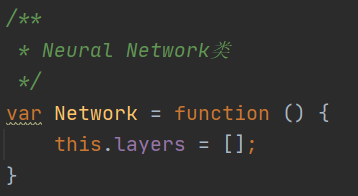
对于每个神经网络层的组成单位--神经元来说，第l层的第j个神经元接收第l-1层所有神经元的输入，对应地存在相应的权值矩阵。在Neuron类的populate方法中对权值进行随机的，在[-1, 1]上的初始化（防止梯度爆炸或更新过慢）。

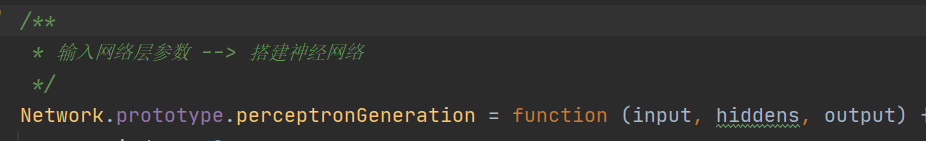
**1.1 构建神经网络层类**

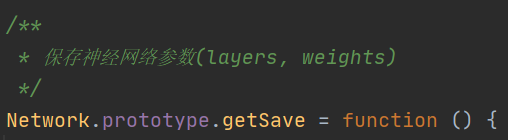


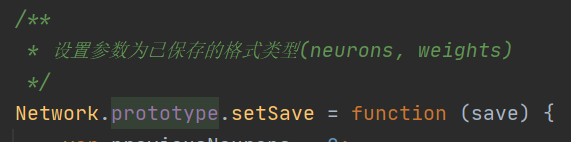
和Neuron类同理，Layer类由一个个Neuron类组成，其中的populate初始化该层的所有神经元与其对应的的权值参数。

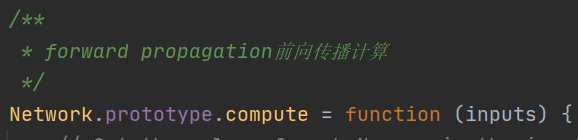
**1.3 构建神经网络类**





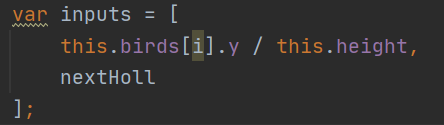




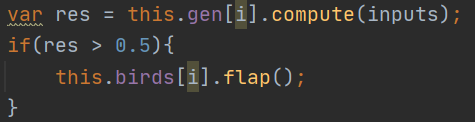


Network类由一层层Layer类构成，其中的方法负责将输入前向传播产生输出(score), 保存以及设置网络参数。

在代码中，采用了两种形式对代码进行存储，一是标准的神经网络结构，其后是神经网络的压缩形式。两种形式可以调用getSave、setSave进行转换。



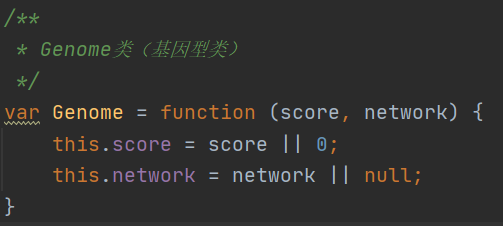
关于神经网络的输入，input\_dim = 2，分别为小鸟的相对位置、下一管道的位置。



神经网络的输出介于0-1之间，是小鸟对是否扇动翅膀进行一次flap进行判断，若输出 > 0.5，小鸟则判定需要进行一次flap以躲避障碍物。

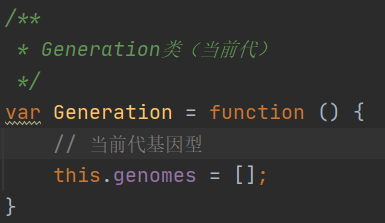
**2、遗传 -- 构建每只小鸟的基因型以及代（Generation）**

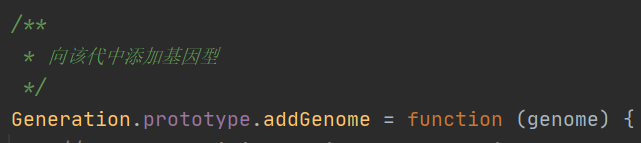
**2.1 构建基因型类**

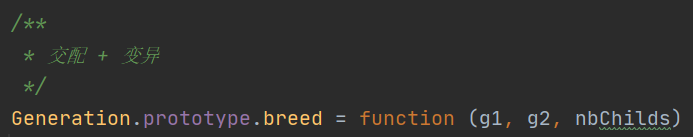


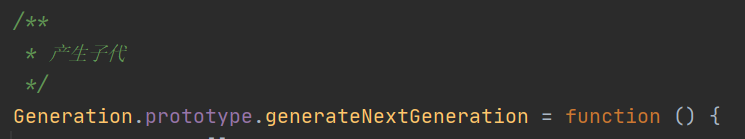
一个Genome类中包含了该基因型的network结构以及对应的score得分。

**2.2 构建世代类**





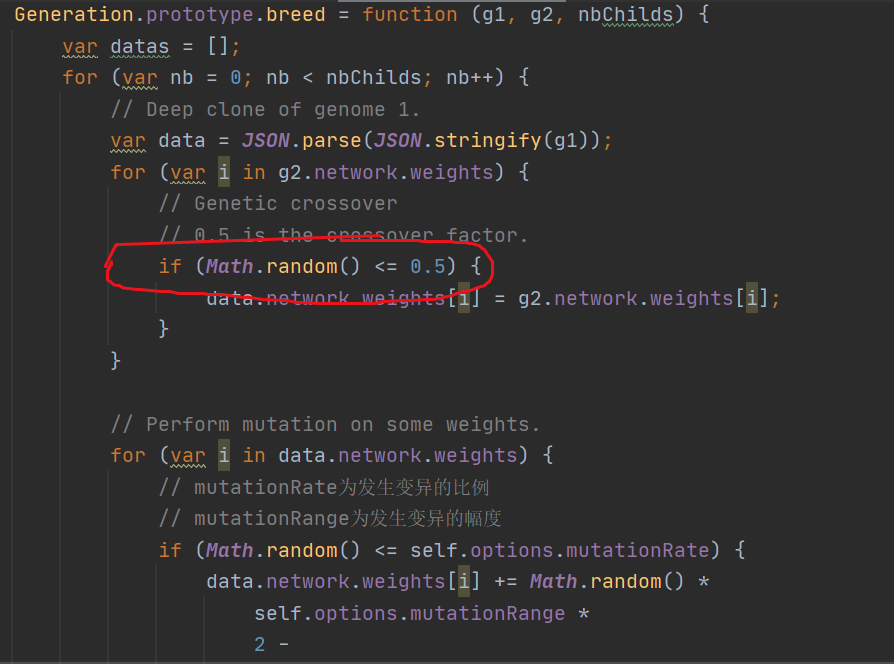




Generation类中提供了添加基因型、交配+变异、繁殖产生子代的方法。

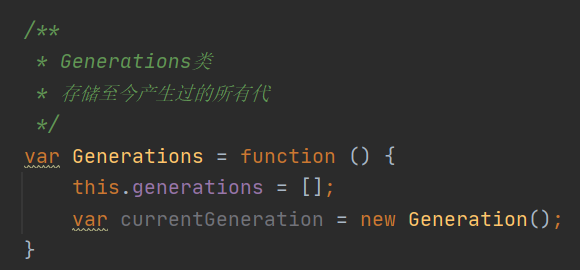
addGenome方法中对所有基因型的得分进行排序，将得分高的基因型排在前面的位置，以便于后续进行选择。

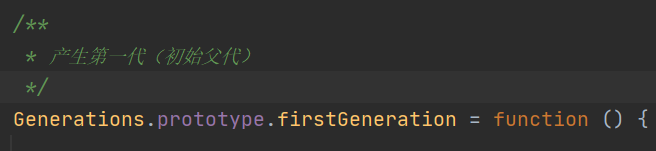
generateNextGeneration中选取这一代较为优质的基因型，使用随机生成一部分基因型并与优质基因型交配，产生新一代的神经网络。

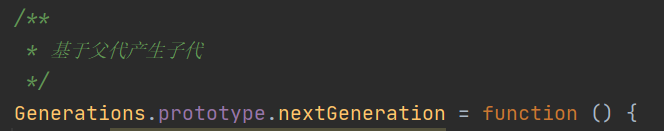


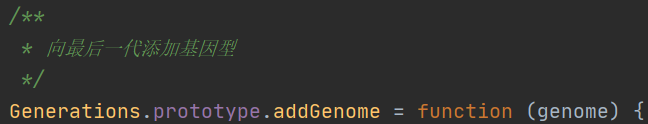
上述breed方法完成了基因突变的一些基本的功能。包括了首先遗传下基因组1和基因组2的一些内容。这里面包含了遗传交叉。遗传交叉就是：有50%的几率遗传基因组1的这些功能，而另外50%就是基因组2的。

**2.3 构建所有世代类**







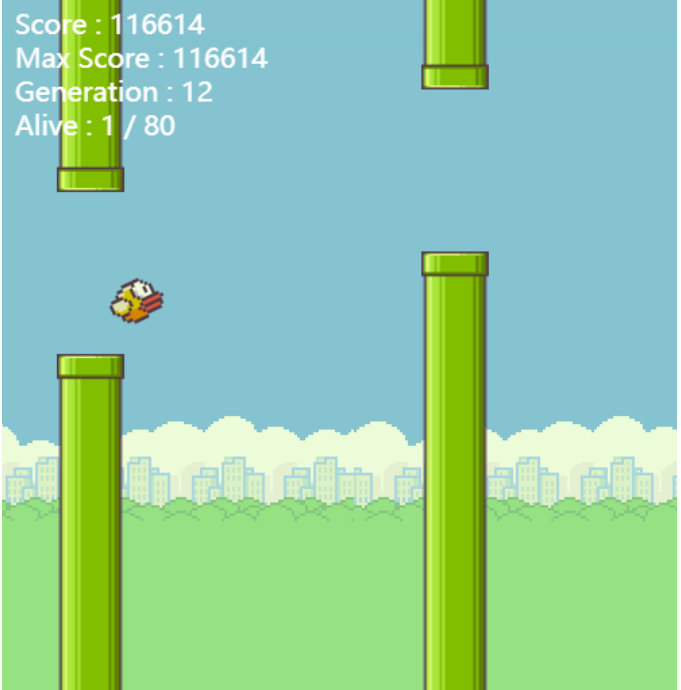


Generations类由Generation类组成，负责迭代产生不同世代，保存最后一代的基因型。

上述算法实现模块只是简要地概括了遗传算法 + 神经网络的大致解决问题的结构与方法。具体代码请见https://github.com/Jiayu-1011/FlappyBirdAI

**五、结果分析与总结**

由于遗传算法的随机性（变异、初始化），每次训练迭代并没有一个固定的迭代次数可以确定地产生出优质的基因型出来，但经过多次试验，大多数情况都可以在50代以内产生很好的效果，score基本都可以达到10万以上。



总结：

神经网络对于游戏问题的策略选择能力无疑是非常强大的，尤其在经过一定的训练之后。由于计算机对参数的学习能力，也就是计算速度要远高于人类，那么使用遗传算法进行迭代筛选后的神经网络更是锦上添花，人类将很难在游戏领域与AI比拟。

本论文仅仅是使用了一个较为简单的机器学习方法对AI玩游戏进行了实现。随着人工智能以及智能计算的进一步发展，AI在游戏领域的统治力将会变得无与伦比。其实游戏，其底层无非就是动态的实时决策和博弈，这也从侧面说明了使用AI智能计算算法，对于决策和博弈能力是一个维度上的飞跃，一下跨越了许多层台阶。