**人工智能实验报告**

目录

[人工智能实验报告 1](#_Toc19461)

[一、总体实验思路 2](#_Toc3072)

[（1）实验目的 2](#_Toc27256)

[（2）实验内容 2](#_Toc24457)

[（3）实验环境 2](#_Toc19415)

[（4）实验方法和合理性分析 2](#_Toc28060)

[二、具体实现 3](#_Toc14514)

[（1）监督学习拍照识别棋子位置 4](#_Toc22770)

[1.实验思路（技术方案） 4](#_Toc31645)

[2.合理性分析 4](#_Toc12679)

[3.实验过程 4](#_Toc32119)

[4.实验结果 5](#_Toc14924)

[5.结果分析 9](#_Toc9424)

[6.可能的改进措施 9](#_Toc14697)

[（2）博弈树搜索实现五子棋博弈 10](#_Toc6562)

[1.实验思路（技术方案） 10](#_Toc3993)

[2.合理性分析 11](#_Toc22461)

[3.实验过程 11](#_Toc30568)

[4.实验结果 11](#_Toc3470)

[5.结果分析 13](#_Toc4955)

[6.可能的改进措施 13](#_Toc29228)

[（3）进化算法训练评估神经网络 13](#_Toc27201)

[1.实验思路（技术方案） 13](#_Toc21718)

[2.合理性分析 14](#_Toc11083)

[3.实验过程 15](#_Toc31040)

[4.实验结果 15](#_Toc8087)

[5.结果分析 16](#_Toc12127)

[6.可能的改进措施 17](#_Toc7501)

[（4）强化学习实现人工智能五子棋程序 17](#_Toc19102)

[1.实验思路（技术方案） 17](#_Toc12427)

[2.合理性分析 18](#_Toc31641)

[3.实验过程 18](#_Toc4260)

[4.实验结果 18](#_Toc26833)

[5.结果分析 19](#_Toc27788)

[6.可能的改进措施 19](#_Toc1537)

[三、实验总结 20](#_Toc12130)

**一、总体实验思路**

**（1）实验目的**

理解神经网络的工作原理，学会自己搭建并用进化算法、强化学习等多种方式训练神经网络，使之能比较好地拟合一些函数，解决一些实际问题。

理解博弈搜索和剪枝的方法，能够运用到AI五子棋的决策中，使AI有一定的五子棋下棋水平。

**（2）实验内容**

实验内容有四项，分别为拍照识别棋子位置、博弈搜索实现五子棋博弈、进化算法训练的神经网络进行五子棋博弈、强化学习训练神经网络实现五子棋博弈，其中拍照识别棋子位置、进化算法训练的神经网络、强化学习三项需要搭建人工神经网络，而第二项问题采用人为设定评估函数的方法，因此无需用神经网络，本质上是对接下来一定步数内的棋局状态的遍历。

**（3）实验环境**

VScode编译环境

Python 3.8.5 64-bit

**（4）实验方法和合理性分析**

为深入理解人工神经网络的建立过程和权值调整方法，做到不光知其然而且知其所以然，本次实验没有选择直接调用库函数建立神经网络，而是选择自己动手搭建三层BP网络，以求深入理解神经网络学习过程。实验要求的1、3、4项，均采用同一框架下的三层BP神经网络。

所用BP网络的整合函数为加权求和函数，激活函数为S型函数的双曲正切函数，基于最小平方误差准则和梯度下降优化方法来确定权值调整公式。误差可以表示为：



其中n表示输出层神经元共n个，表示第i个输出层神经元的目标值，表示第i个神经元的实际输出值。

梯度下降法调节误差的公式为：



推导出隐含层-输出层之间的权值调整公式为：



其中为第j个隐含层神经元与第i个输出层神经元之间的连接权值，η为学习率。

输入层-隐含层之间的权值调整公式为：



其中表示输入层第k个神经元与隐含层第j个神经元之间的权值，为第k个神经元传给隐含层第j个神经元的值，表示第j个隐含层神经元的输出值。

具体实验中创建一个BPNet类，将BP网络封装起来使用，可以通过输入值决定BP网络各层有多少个神经元、学习率、迭代次数等。

合理性分析：BP神经网络具有以下优点：

1) 非线性映射能力：BP神经网络实质上实现了一个从输入到输出的映射功能，数学理论证明三层的前馈神经网络就能够以任意精度逼近任何非线性连续函数，这使得其特别适合于求解内部机制复杂的问题，即BP神经网络具有较强的非线性映射能力，因此可用于模拟图片和棋子位置之间的映射关系，也可以模拟棋盘状态和对此状态下每个动作的收益值的映射关系。

2) 自学习和自适应能力：BP神经网络在训练时，能够通过学习自动提取输出、输出数据间的“合理规则”，并自适应的将学习内容记忆于网络的权值中。即BP神经网络具有高度自学习和自适应的能力。

3) 泛化能力：在设计模式分类器时，不仅要考虑网络在保证对所需处理对象进行正确处理，还要关心网络在经过训练后，能否对未见过的模式或有噪声污染的模式，进行正确的处理。也即BP神经网络具有将学习成果应用于新知识的能力。

4) 容错能力：BP神经网络在其局部的或者部分的神经元受到破坏后对全局的训练结果不会造成很大的影响，也就是说即使系统在受到局部损伤时还是可以正常工作的。即BP神经网络具有一定的容错能力。

综上，可以认为BP网络有解决实验要求的能力，这使得实验总体合理性得到初步保证，对每个要求所采用的具体实现方法，接下来还会具体分析其科学性和合理性。

**二、具体实现**

**（1）监督学习拍照识别棋子位置**

**1.实验思路（技术方案）**

确定实验思路和技术方案前，应先了解计算机如何处理图片。在计算机视角下，图片也是二进制的编码。图片的内容完全由其包含的所有像素点的颜色属性确定，每个像素点在RGB颜色模式下可以表示为为一个一维向量(x,y,z)，三个分量分别表示三种颜色的强度。一张图片的所有像素向量就可以作为这个图片的特征描述向量进行计算。

为进一步提高程序处理图片的效率，还可以对图片进行灰化处理，此时图片的每个像素点向量只有一个分量，表示该位置的黑白程度。这样，对于一个M\*N像素大小的棋盘，就可以用一个M\*N的矩阵表示。且五子棋子颜色只有黑和白，因此灰化也能一定程度降低图片中其他无关内容的干扰。

能够读入图片特征向量后，就能把特征向量作为BP网络的输入，而棋子位置为神经网络的输出。实验图片像素大小为300\*300，因此输入层有90000个神经元，图片内容为为19\*19的棋盘，如图1所示，因此输出为一个19\*19的矩阵，这也意味着输出层神经元应有361个。用1和-1分别表示己方和敌方棋子，没有棋子用0标识。隐含层神经元数量可在实验中根据程序运行效果调整。

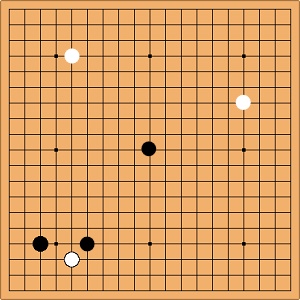


图1

**2.合理性分析**

上文曾经分析过，BP网络有良好的非线性映射能力以及泛化能力，三层的BP网络即可以任意精度逼近任何非线性连续函数，适合于求解内部机制复杂的问题。第一项实验就是要完成由图像像素矩阵到棋盘矩阵的映射，因此可以采用BP网络运算。

**3.实验过程**

实验数据为两张300\*300像素的.jpg图片，如图2、3所示。调用cv2.imread(filepath)函数将照片读入，调用cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)函数对图片做灰化处理，方便数据输入。对于传入的图片，需要人工确定其真正对应的棋盘状态，作为神经网络监督学习需要训练的目标值。调用根据上文已经编写好的BPNet类生成BP神经网络。对每个神经元输出，如果为负值且明显小于0，则视为-1，即对方棋子；如果为正且明显大于零，则视为1，即己方棋子。分别将隐含层节点设置为3个、10个、15个，以便明确隐含层神经元数量对程序运行时间和精确度的关系。以上情况均需要运行程序直到神经网络误差收敛。最后为测试训练好的神经网络，传入一张陌生图片检验训练效果，陌生图片为图4。

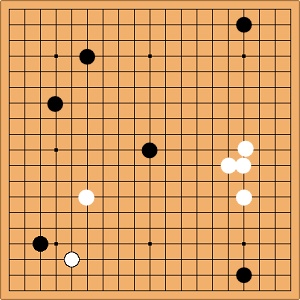
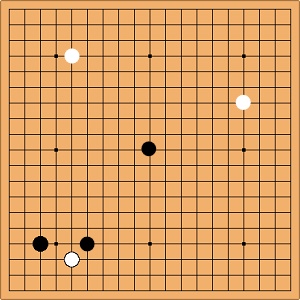
 

图2 图3

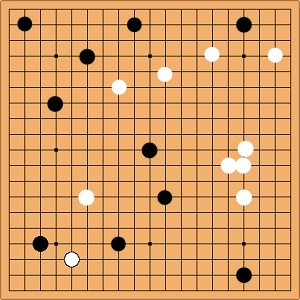


图4

**4.实验结果**

隐含层含3个神经元时，训练至误差收敛后，对于两张输入图片，第一张图片拟合得比较好，与原图片对应的棋盘状态一致，但对第二张图片拟合效果不好，具体表现为第二张图片的输出与第一张图片的输出变化不大，尽管这两张图片对用的棋盘状态有一定差距。而且测试图片也和图一输出基本一致。误差和输出如下图：

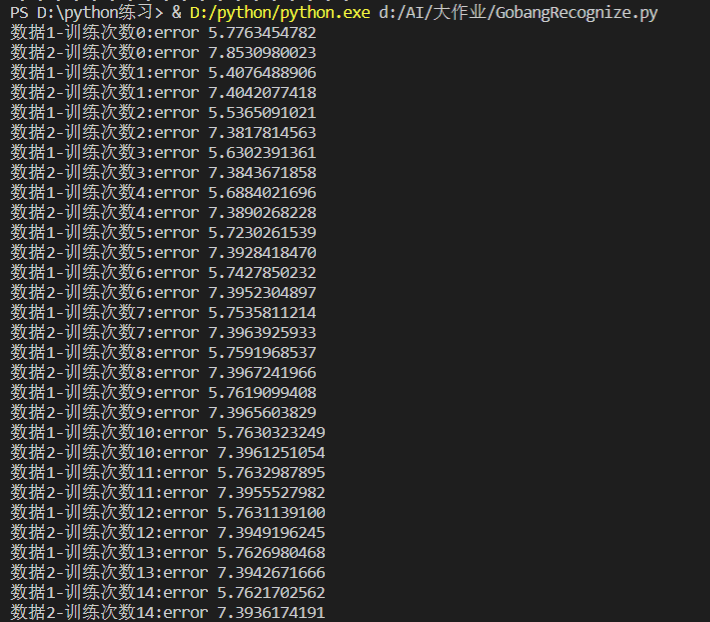


图5-误差

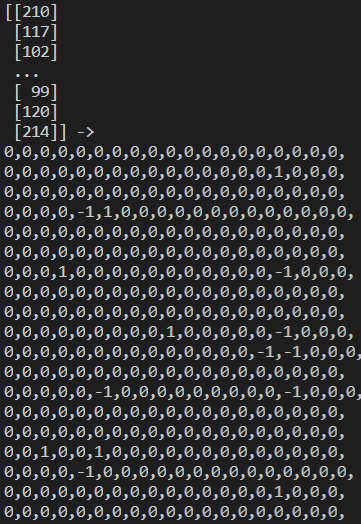
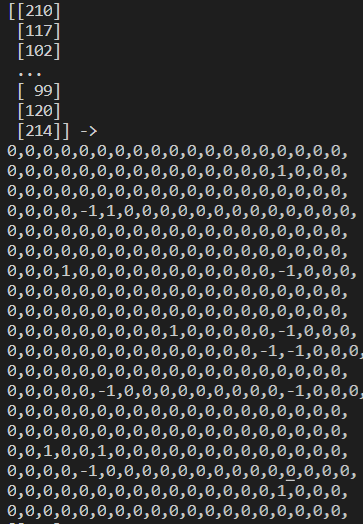
 

图6-训练输入1及输出1 图7-训练输入2及输出2

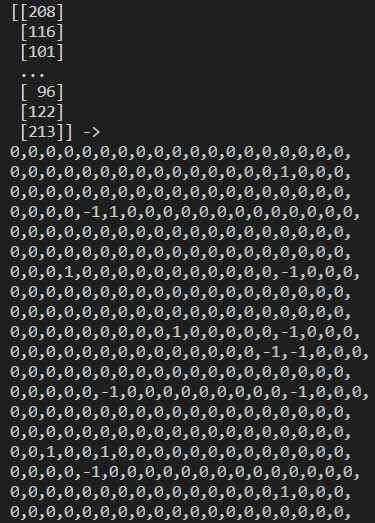


图8-测试输入及输出

隐含层含10个神经元时，网络能够区分输入1和输入2，对这两个数据的输出均与目标输出一致，对于测试输入，网络已经能区分其与训练数据的差别，但效果还不是很准确。同时由于网络结构变得复杂一些，训练过程变得比较慢。输出结果如下：

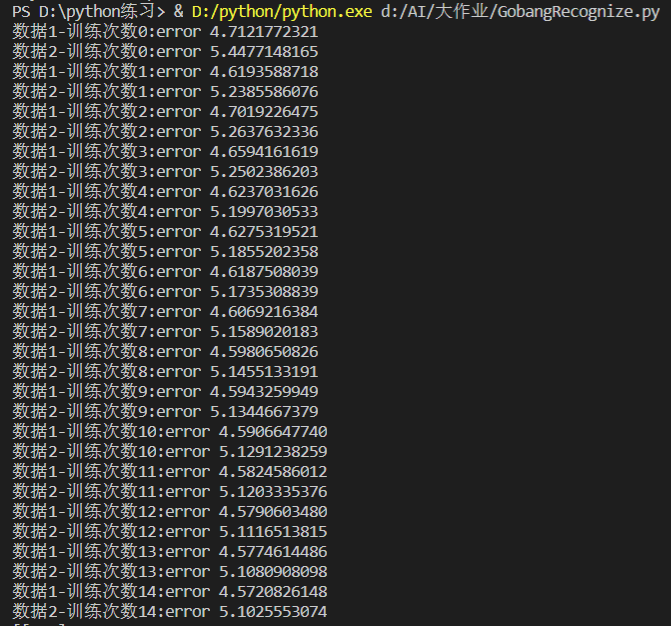
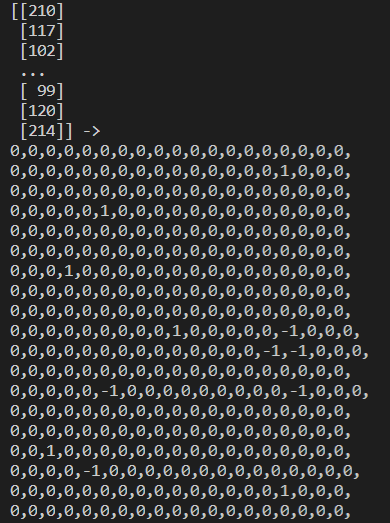
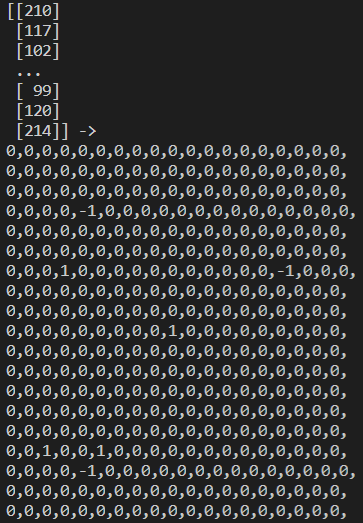


图9-误差

图十10-训练输入1及输出1 图11-训练输入2及输出2

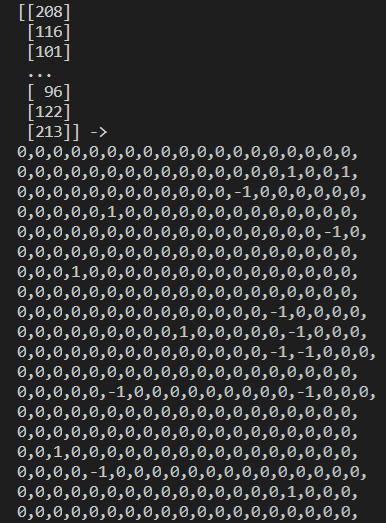


图12-测试输入及输出

隐含层含15个神经元时，多次训练后神经网络并没有收敛（已通过将权值保存在文件中进行了多次训练），且训练速度更慢。对于输入1、输入2和测试输入均没有更好效果。结果如下：

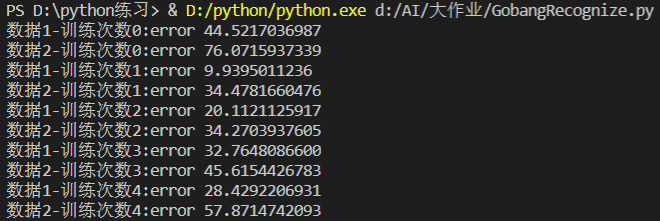


图13-误差

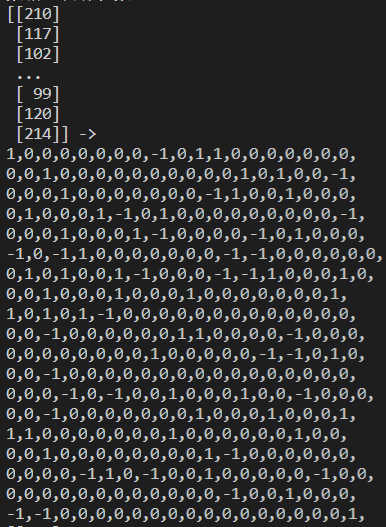
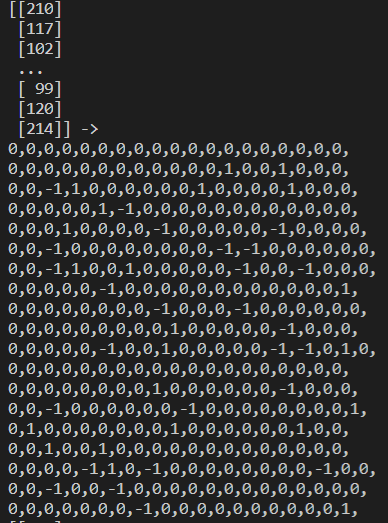
 

图14-训练输入1及输出1 图15-训练输入2及输出2

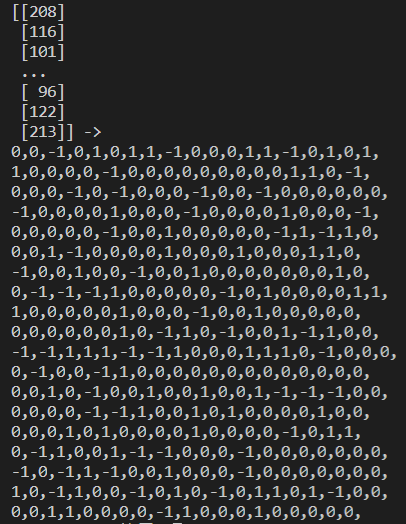


图16-测试输入及输出

**5.结果分析**

观察图片像素特征向量可以得知，灰化后的棋盘在没有棋子的地方，仍有一定值，是介于黑和白之间的颜色值，而由于棋子面积相对于棋盘面积来说比较小，因此传入的图片数据很大一部分是相同的。当隐含层有3个神经元时，输出情况说明当前网络结构还比较简单，还不能根据输入区分不同图片。这就对网络结构提出了更高的要求。隐含层含10个神经元时，网络能够对输入数据做很好的区分，但存在过拟合的问题，对陌生数据表现不佳。当隐含层含15个神经元时，网络结构变得更复杂，但效果反而下降，甚至不如隐含层为3个神经元时的情况。查阅资料后得知，如果隐层结点数过少，网络不能具有必要的学习能力和信息预测处理能力。反之，若过多，不仅会大大增加网络结构的复杂性（这一点对硬件实现的网络尤其重要），网络在学习过程中更易陷入局部极小点，而且会使网络的学习速度变得很慢。因此得到的一个结论是隐含层为10个神经元是比较适合本问题的。同时，网络结构毕竟仅有三层，无法对更复杂的问题作出很好的结果。

结合所学和以上实验结果可以看出BP网络在实际应用中存在以下缺陷：

1.BP算法为一种局部搜索的优化方法，但它要解决的问题为求解复杂非线性函数的全局极值，因此，算法很有可能陷入局部极值，使训练失败；

2.网络的泛化能力与逼近能力的矛盾。一般情况下，训练能力差时，预测能力也差，并且一定程度上，随训练能力地提高，预测能力也提高。但这种趋势有一个极限，当达到此极限时，随训练能力的提高，预测能力反而下降，即出现所谓“过拟合”现象。此时，网络学习了过多的样本细节，而不能反映样本内含的规律，对陌生数据表现很差；

3.网络结构的选择尚无一种统一而完整的理论指导，一般只能由经验选定。

**6.可能的改进措施**

以上实验结果并未达到预期，主要原因就在于数据量不够、最优网络结构难以确定、网络收敛慢和BP网络容易出现过拟合等问题。

可能的优化办法：

1.自适应调节学习率。初始时，学习率设置成一个比较大的数，如可以设置为0.1。随着训练的进行，网络误差逐渐减小，需要更小的移动量，此时可以把学习率调低，以此加快收敛速度。

2.选取更多的隐含层神经元数量进行测试，挑选出其中最适合本问题的隐含层结构。

3.Early stopping方法。为防止过拟合问题，每次对训练集训练完后，就计算网络的误差值的变化量，当误差变化量不再明显时就退出训练。

4.加大数据量。由于人工确定一个照片对应的棋盘状态比较费时间，因此实验没有大量数据可用。如果能自动生成图片和目标值，则能解决这一问题。

**（2）博弈树搜索实现五子棋博弈**

**1.实验思路（技术方案）**

博弈搜索让计算机把每一步的可能性都试一遍，看走在哪效果最好。实验采用15\*15棋盘。对于一个棋局，判断它对自己来说是占优势还是劣势，用人为确定的数值来评估。对于五子棋就是统计目前的棋型，并累加分数。比如如果有4个子连起来了，那就给个很高的评分，因为下一步就可以赢了，如果是3个子连起来了，给个相对较低的评分，因为不一定就能赢，对方可能会及时封堵，但是仍然比只有2 个子连在一起的得分要高，用下面的棋型评分表判断得分：

shape\_score=[(50\*1e-9,(0,1,1,0,0)),

(50\*1e-9,(0,0,1,1,0)),

(200\*1e-9,(1,1,0,1,0)),

(500\*1e-9,(0,0,1,1,1)),

(500\*1e-9,(1,1,1,0,0)),

(5000\*1e-9,(0,1,1,1,0)),

(5000\*1e-9,(0,1,0,1,1,0)),

(5000\*1e-9,(0,1,1,0,1,0)),

(5000\*1e-9,(1,1,1,0,1)),

(5000\*1e-9,(1,1,0,1,1)),

(5000\*1e-9,(1,0,1,1,1)),

(5000\*1e-9,(1,1,1,1,0)),

(5000\*1e-9,(0,1,1,1,1)),

(5000\*1e-9,(0,1,1,1,1,0)),

(1e9\*1e-9,(1,1,1,1,1))]

同第一问，1表示己方棋子，0表示无棋子。

要计算某一棋局状态对自己的分数，需要统计所有匹配的棋型得分并累加。为此需要编写一评估函数。本实验采用的评估函数按照上面的评分表计算分数，对某一形状中的棋子，只计算该形状得分一次，不会对每个棋子都计算一次。如果有得分形状交叉，则得分将翻倍提高。

AI接下来要走的位置，就是计算它在走某一个点后，计算局面的得分，然后得分最大的那个点。

如果要使AI能有较高的博弈水平，就需要使其有一定预判能力，因此仅仅是贪心选择当前一步内收益最大的位置是不够的，需要搜索更深层次的棋局状态。过程中AI将判断棋局对应的对手的分数，以便能选择使对手收益最小的位置。

但搜索深度提高将使程序时间复杂度呈指数增长。为保证程序有较快的反应速度，需要进行α-β剪枝。

博弈过程中，双方均会选择对自己来说的最优方案，即评估值为MAX，作为对手来说，要选择的是对对方评估值最小的为MIN。为每个节点设立α值和β值，初始α=−∞，β=+∞，表示一个节点的值范围，通过这两个值的向下传递和向上更新来完成隔代剪枝。如果以MAX节点表示己方决策节点，以MIN节点表示对方决策节点，则当任何MIN顶点的β值小于或等于其某个祖先MAX节点的α值时，MIN节点以下分支可停止搜索，并令MIN节点倒推值为β，此时发生α剪枝；任何MAX顶点的α值大于或等于其某个祖先MIN顶点的β值时，MAX顶点以下分支可停止搜索，并令MAX节点倒推值为α，此时发生β剪枝。

博弈双发都站在自己角度评估当前棋局下自己的分数，根据这个特点，可以把对方下一步的得分值的相反数作为自己当前这一步的评分，这就是负值极大法。这样做可以使双方使用同一个评估函数。

为方便三、四问调用，整个博弈树及其操作作为一个类封装，同时提供人机对抗和AI互相对抗两种运行方式。

**2.合理性分析**

评分表人为设定，之所以需要乘1e-9是因为后两问将棋盘状态作为神经网络输入，而棋局评分作为输出，因此棋局评分不宜过大，否则权值调整时学习率将不容易确定。从评分表也可看出一旦出现己方连线被对手封堵后，此未完成的连线将不再对自己有加分，符合常理。

对于有得分形状交叉的情况，评分应该有额外加成，因为此时对方封堵将变得困难，符合实际情况。

博弈树搜索本质是对接下来所有情况的遍历，因此其实现情况稳定且效果明显。搜索深度不宜过深，否则计算机将计算很长时间才能有结果，一般搜索深度为3。采用α-β剪枝能避免搜索不必要路径浪费时间，提高搜索速度，使AI比较快的反应能力。

**3.实验过程**

规定用“-”表示AI1或对方棋子，用“+”表示AI2或己方棋子。AI互相对抗时，AI2先落子；人机对抗时，人类先落子。在搜索深度为1和3的情况下分别以AI互相对抗的方式运行程序，观察程序执行效果。接着以人机对抗的方式运行程序，检验AI五子棋下棋水平。

**4.实验结果**

搜索深度为1时，AI反应迅速，AI互相对抗的双方能对局势做初步判断。棋局结束时棋盘状态如图17：

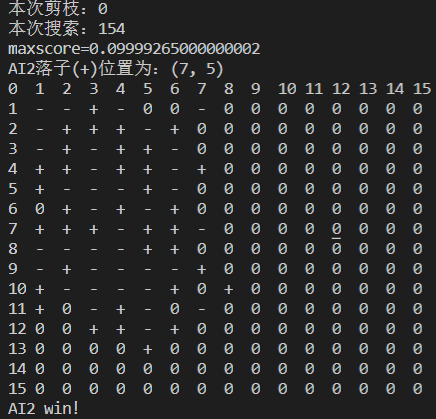


图17

其中“-”表示AI1，“+”表示AI2。

人机对抗的结果如图18：

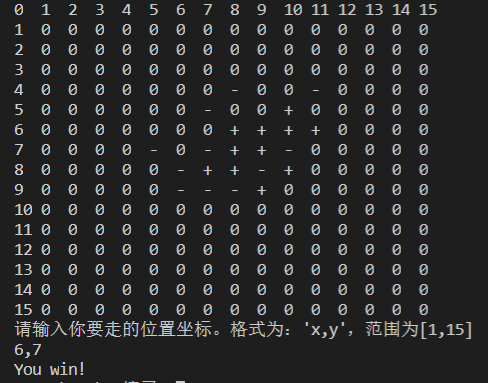


图18

下棋时，发现AI已经有一定的下棋水平。从结束时棋盘状态也能看出，经常有四子连线被堵住的情况，说明AI可以对一些情况做出及时封堵。

搜索深度为3时，AI对抗结果如下图：

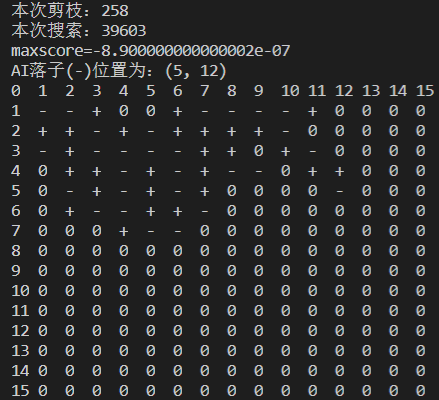


图19

由于此时搜索已经非常耗时，所以很长时间后棋局仍然没有结束。仅展示以上未完成的棋盘。

人机对抗结果如下图：

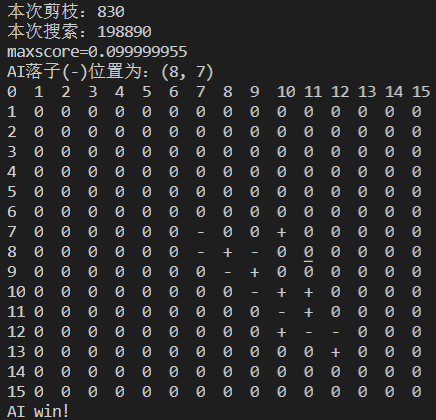


图20

此时AI已经有比较高的下棋水平，即使对多个形状交叉的情况也能做出正确判断，要战胜有一定难度。

**5.结果分析**

搜索深度为1时，AI有一定下棋水平，反应快，但对于多个形状交叉时的情况无法做出准确判断，因此被战胜还比较容易。

搜索深度为3时，AI已经有了比较高的下棋水平，达到了预期水平，说明选取的评估函数比较合理。因为其能遍历接下来3步内的棋盘情况，对棋盘大部分情况均能做出比较聪明的判断。但代价是运算速度变得比较慢，因为此时搜索量已经呈指数级增长。

**6.可能的改进措施**

博弈树搜索采用遍历棋盘情况的方法，因此其实现结果比较好，在评估函数合理的情况下AI能有很好的下棋水平，但缺点是需要以巨大的计算量为代价，搜索深度一旦较深出结果就会很慢。因此主要改进方向有两方面，一是采用更合理的评估函数以更好预测局势走向，二是除了采用α-β剪枝还要采用其他辅助方法缩小搜索范围，提高运算速度。

更合理的评估函数主要是修改各个形状的得分，或者改变对多形状交叉的评分方法。但对评估函数的修改没有具体的指导方法，且由于本实验结果显示AI已经有了比较高的下棋水平，因此保持原来的评分方法，不必再花大量时间做改变。

实验采用的方法对搜索范围的缩小还有提升空间，可以结合常识对搜索范围做约束。比如，一般来说，自己下一步棋一般在上次落子附近，可以根据这个规律对棋盘搜索空间做一个排序，以此加快搜索速度。

**（3）进化算法训练评估神经网络**

**1.实验思路（技术方案）**

首先需要明确神经网络在博弈中的作用。在本问题中棋局状态作为神经网络的输入，对棋局的评分为输出，以此替代人为设定的评估函数。实验采用遗传算法训练和优化BP网络，一方面减小误差，另一方面加快收敛速度。

遗传算法是模拟自然界遗传机制和生物进化论而成的一种并行随机搜索最优化方法。基于自然界“优胜劣汰，适者生存”的生物进化原理引入优化参数形成的编码串联群体中，按照所选择的适应度函数并通过遗传中的选择、交叉和变异对个体进行筛选，使适应度值好的个体被保留，适应度差的个体被淘汰。本问题中，可以把第二问中某一棋局状态和其对应的评估分数作为训练数据输入，随机生成多个神经网络，对这些网络做进化计算。适应度函数与网络计算误差联系，误差越小说明个体适应度越大，因此适应度函数可以取误差的倒数。新的群体既继承了上一代的信息，又优于上一代。反复循环，直至网络误差较低。算法流程图如下图：

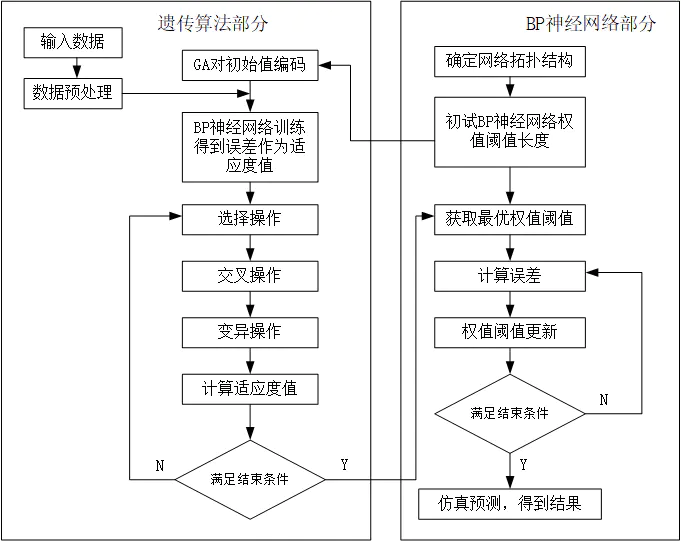


图21

遗传算法优化BP神经网络的目的是通过遗传算法得到更好的网络初始权值和阈值，其基本思想就是用个体代表网络的初始权值和阈值、个体值初始化的BP神经网络的预测误差作为该个体的适应度值，通过选择、交叉、变异操作寻找最优个体，即最优的BP神经网络初始权值。

种群中的每个个体是问题的一个解。在本问题中对应某一权值下的一个神经网络。网络的各个权值可以作为“染色体”，也是这个网络的所有可遗传信息。如果采用浮点数编码，一个权值就是网络的一个基因。通过遗传可以把较优网络的权值传给下一代。

遗传算法的基本操做有选择操作、交叉操作、变异操作。本实验选择操作采用轮盘赌方式实现，交叉操作将交换两个网络之间的某一权值，变异操作由一个变异几率参数控制，使网络有一定探索能力，能跳出局部最优点。

**2.合理性分析**

遗传算法是具有高效启发式搜索、并行计算等特点，应用于函数优化、组合优化及生产调度等方面。用遗传算法优化BP神经网络的初始权值和阈值，使优化后的BP神经网络能够更好的预测函数输出。同时一般的迭代方法容易陷入局部极小的陷阱而出现死循环现象，使迭代无法进行。遗传算法很好地克服了这个缺点，是一种全局优化算法。  
 **本问题中遗传算法基本要素包括**染色体编码方法、适应度函数、遗传操作（选择、交叉、变异）、运行参数（参数：群体大小、遗传代数、交叉概率和变异概率）四个方面，能实现遗传算法的全部过程。

**3.实验过程**

首先确定遗传代数，即设定算法迭代次数。之后确定种群数量、遗传变异几率等。本实验中种群规模设置为40，交叉率为0.7，变异率为0.01。

然后通过人机对抗或AI互相对抗的方式，获取到一盘棋中的训练数据，包括当前棋局状态和己方对这一状态的评分，这可以通过运行第二问程序得到。

调用写好的BP网络类，创建一个输入层为225个神经元、隐含层有3个神经元、输出层为1个神经元的网络，用它计算某一棋局对己方的评分。

获取到训练数据后，就可以开始进化计算。为使效果更加明显，将种群平均误差曲线和最优个体误差值画出图像。

进化计算完成后，将最优个体情况记录下来，为进一步缩小误差，还可以对其进行监督学习训练。

**4.实验结果**

训练200代后误差情况如图：

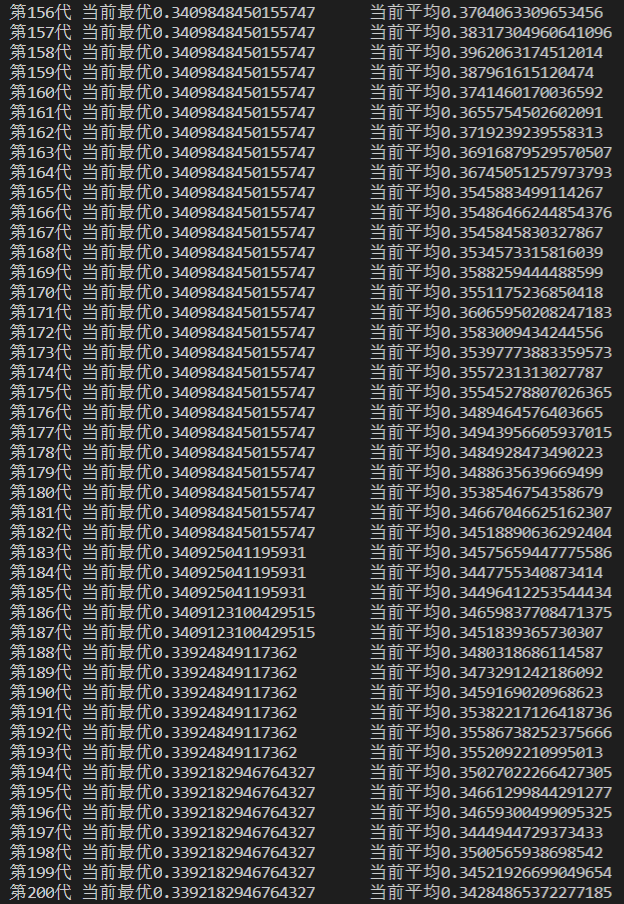


图22

从最后得到的误差图像中可以看出遗传算法取得了比较好的降低误差的效果。图中红线为最优个体误差，蓝线为种群平均误差。

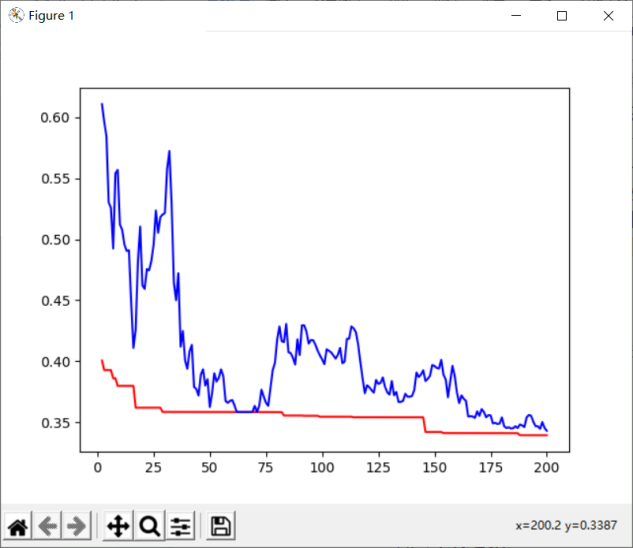


图23

将训练好的网络用于人机对抗，棋局结束时的状态为：

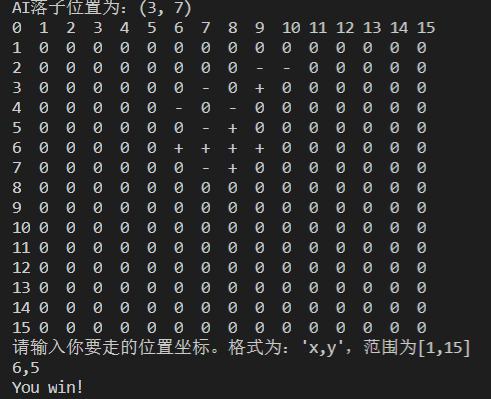


图24

对局结果显示，AI有一定的意识使自己的棋子尽量集中且靠近人类棋子，与随机落子有明显区别，但还无法对人类的棋子做出封堵。

**5.结果分析**

实验结果显示，用遗传算法能明显降低网络计算误差，具有良好训练效果，达到了降低误差、优化网络的作用。

但误差降低效果显著不代表AI有很强的下棋水平。训练完后发现AI下棋水平不高，仅仅有在人类落子附近落子的意识，而无法对人类行动做出准确封堵。

第二问时，采用相同评估函数，AI在搜索深度为3时能有很好的下棋水平，说明本次实验的训练数据的正确性是有保证的，网络降低误差的效果也比较明显，那么问题很可能是训练数据量太小的问题。对于一个15\*15的棋盘，棋局的状态空间容量为3^225，是个相当大的数量。通过神经网络判断当前局势，可以想象对训练数据的需求量也是非常巨大的，需要考虑众多情况，实验中使用的数据量是远远不够的，远远达不到可以使AI发现下棋规律的程度，无法做到对许许多多情况下的训练，因此效果比较差。

**6.可能的改进措施**

改进可以从两方面考虑，一是优化网络结构，二是优化进化计算方法，三是加大训练数据量。其中加大数据量是最应考虑的选项。

实验中人为选取了BP网络的结构，很可能不是最适合这个问题的网络。尽管采取进化算法进行训练明显降低了误差，但仍不能保证网络是全局最优。同时，上文提到过，对隐含层的结构没有具体的指导方法，不断测试最优的网络结构需要花费大量时间，因此不方便在这方面做出改进。

遗传算法中交叉、变异、选择的实现有许多参数,如交叉率和变异率,并且这些参数的选择严重影响解的品质,但目前这些参数的选择大部分是依靠经验，这也是遗传算法的缺点之一。因此从这一方面改进提升有限，无法系统进行。

加大数据量是相对来说改进方向明确的方法。要使AI能从实际棋局状态中归纳出下棋规律，需要巨大数据量的支持。本实验中无法满足这一需求，导致效果不佳。网络可以快速收敛，具有比较强的误差调节能力，因此认为投入大量数据后，能训练出比较好的下棋水平。

**（4）强化学习实现人工智能五子棋程序**

**1.实验思路（技术方案）**

强化学习作为训练神经网络的一种方法，根据外界对网络采取的某一行动（action）的奖励（reward）来训练网络，直到网络自己就能判断某一行动对自己的收益。强化学习既不属于监督学习，也不属于非监督学习，奖励值和监督学习的输出值不一样，它不是事先给出的，而是延后给出的。要注意的是每个行动的收益不光包括外界对网络的奖励，还应考虑到采取这一行动后对自己的长远收益，这可以通过计算下一状态时自己能获得的最大收益得到。

强化学习算法已经考虑到未来的收益情况，因此第四问中不宜再使用三层博弈树的搜索策略，否则即使AI下棋效果好也不能说明是强化学习的作用。因此这里相当于只用一层博弈树搜索，选择接下来一步使自己收益最大的情况即可。

当问题状态空间数量不大时，用一张表（称为Q表）就能存储从某一状态到下一状态对应的收益值，这种方法叫做Q-Learning方法。但对本题3^225的状态空间，就必须采取用近似方法评估收益值。神经网络可以做到对收益函数的拟合，是对Q表的近似。

本实验采用最基本的nips-DQN算法，将传统的强化学习算法Q-Learning与神经网络相结合，将状态向量作为神经网络的输入，输出得到相应状态每一个动作的价值。

DQN主要使用的技巧是经验回放（experience replay）,即将每次和环境交互得到的奖励与状态更新情况都保存到经验回放池中，用于后面目标Q值的更新。对于Q-Learning方法，它是有一张Q表来保存所有的Q值的当前结果的，但是DQN是没有的，在做动作价值函数更新的时候，需要其他的方法，这个方法就是经验回放。

为避免算法陷入局部最优，同遗传算法一样添加一些随机性，算法有几率随机选择一个行动。

通过经验回放得到的目标Q值和通过Q网络计算的Q值是有误差的，可以通过梯度的反向传播来更新神经网络的参数，当收敛后，就得到近似的Q值计算方法。算法流程如下图：

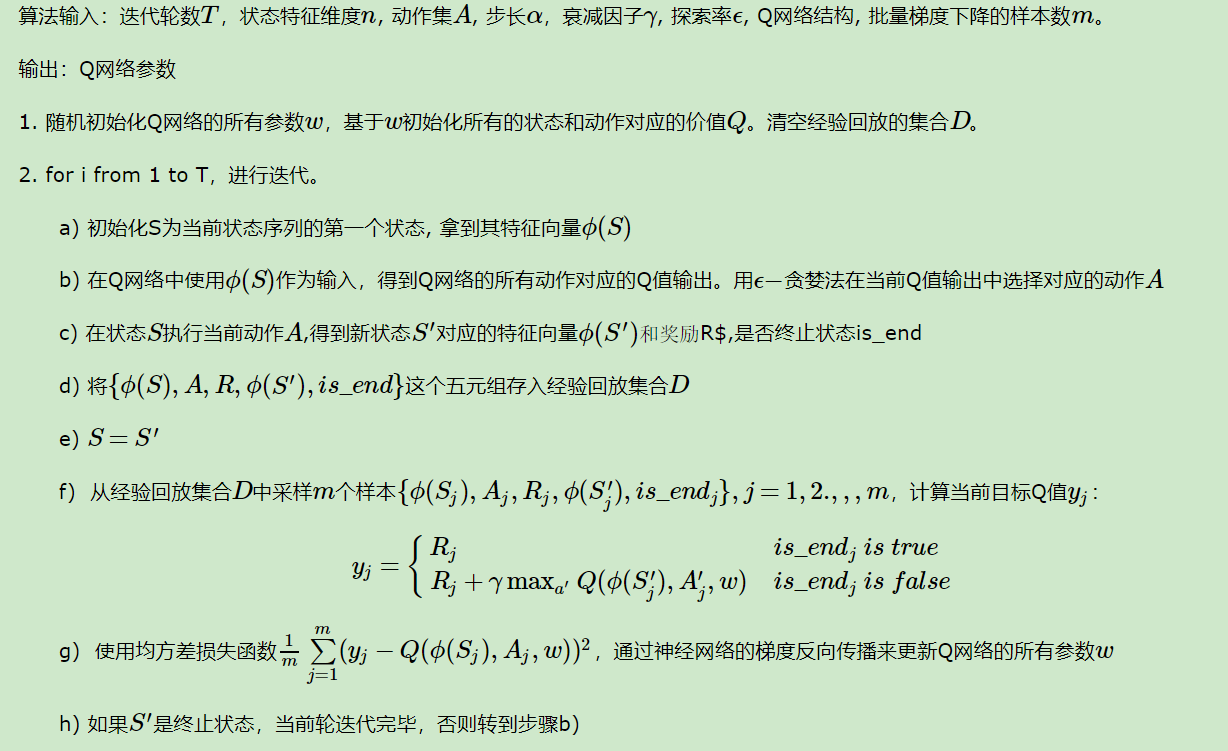


图25

**2.合理性分析**

强化学习的学习思路和人比较类似，是在实践中学习，具有普适性，因此在许多领域都有研究，例如博弈论、控制论、运筹学、信息论、模拟优化方法、多主体系统学习、群体智能、统计学以及遗传算法。同时对于本问题状态空间巨大的情况，采用神经网络评估棋局分数也是强化学习中应用广泛的估计方法。这就给本问题的实现提供了理论支撑，具有可实现性。

**3.实验过程**

实验需要先创建一个神经网络，作为评估某个行动得分的网络。

然后调用第二问中的博弈树搜索AI，与依靠神经网络评分的AI下棋，并将神经网络AI的每一步，连同这一步的收益、当前状态、下一步状态、棋局是否结束作为一组经验放到经验回放池中，其中奖励值是人为设定的，本实验采用了这样的奖励机制：如果这一步神经网络赢，则奖励值为1；若输则为-1；若既没有赢也没有输则奖励1e-9，这是因为考虑到第二问中AI具有较高的下棋水平，通过每多走一步就能获取更多奖励的方式，激励神经网络AI能尽可能多得与第二问中AI博弈。

如果经验池有了一定数量的经验组，就可以开始对网络的训练。计算当前状态下的最大收益，之后的训练公式已在方案中给出。其中下一状态时的最大可能获得的收益由神经网络得到。

每训练100次，就进行人机对抗检验训练效果。

**4.实验结果**

未经过训练的网络对抗结果：

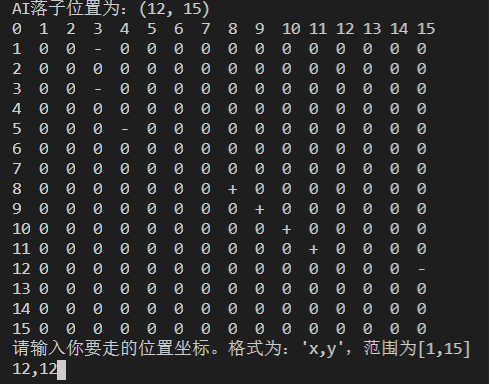


图26

训练100次后的效果：

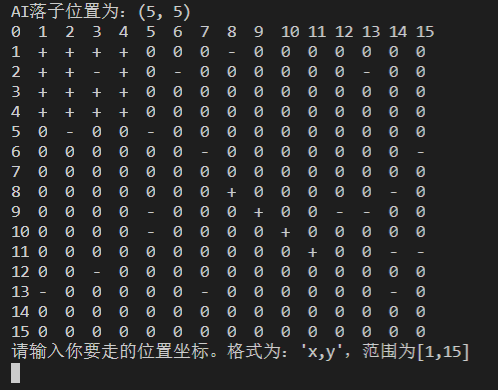


图27

为使AI能多走几步以便观察具体训练效果，没有一开始就下满5颗棋子连线。而此时网络误差显示，算法已经收敛且误差很小。继续训练不会有更明显的提升。

**5.结果分析**

未经训练时，AI行动完全随机。训练后，AI下棋水平有略微提升。具体表现为能有目的地将自己棋子形成2个或3个连线的形状，但由于还不会封堵对手棋子，因此也很容易被打败。

算法没有实现预期水平，问题可能出在对训练数据的选取和对奖励机制的设定上。本问题采用和第二问AI博弈的方式获取训练数据，而第二问有比较高的下棋水平，导致神经网络一直输，从而拿不到获胜时的奖励，因此网络可能就不会朝着五子连线的方向训练。另外每次行动的奖励方法采用的是人为规定的，导致训练方向有一定不确定性，不能保证这样的奖励机制很符合问题的解决。

**6.可能的改进措施**

结合以上分析，本问题可以从网络结构、奖励机制、数据选取方面做优化。

首先网络结构，同第三问，对隐含层的结构没有具体的指导方法，需要借助大量训练和人为观察。

奖励机制方面，由于真实情况下对某一行动的收益和其以后状态集中的最大收益评估存在难度，因此也只能人为设想比较合理的评分方法，同样需要大量人工试验。

改进数据集的选取，因为第二问实现效果较好，神经网络AI在训练时始终处于劣势，无法得到最大程度的正向奖励，因此训练出来不会有很强的连线意识。可以在训练初期选用较弱的人工智能，随着训练的进行训练中使用的对手AI强度才会提升，保证能充分利用训练数据，使AI训练更加全面。

**三、实验总结**

本次实验应用了人工智能中的诸多方法，有BP网络、博弈树搜索和α-β剪枝、遗传算法、强化学习，完成了实验要求。

实验中第二项内容没有采用神经网络方法，其他三项均用自己搭建的三层BP网络实现。实验效果表明，第二项内容实现很好，AI已经有了很强的下棋水平，而其他三项实验效果没有达到预期。分析原因可能是神经网络结构问题，以及数据量太小的问题，这也反映了一些神经网络本身的缺点。

第二问采用博弈树搜索，是完全确定的算法，只要评估函数合理，实验结果就能得到保证。其他三问用的神经网络算法有一个广为人知的不足——黑盒子性质。神经网络的实质是对一个不明朗的数学函数的拟合，不管神经网络训练结果好坏，其内部权值调整的方向是很难解释和预测的。即使有相关领域的经验，训练一个好的神经网络往往也需要大量的数据支撑，从而对计算机运算速度有很高的要求。本次实验中，对五子棋棋局评分没有一个绝对准确的方法，只是依靠经验，给神经网络的训练带来不确定性；同时数据量不足以及个人计算机运算速度比较慢，使网络没有得到充分训练。

未来的改进方向，一是寻找更合理的网络结构。受限于个人水平，目前只有三层网络，结构还比较简单；二是编写一些程序自动生成大量数据，使神经网络得到充分训练，提高博弈水平。