|  |
| --- |
| 电子科技大学  **计算机专业类课程** |
| **实验报告** |
| **课程名称：高级程序设计**  **学院专业：计算机科学与工程学院**  **学生姓名：方彦嘉**  **学　　号：2023080905003**  **指导教师：俸志刚** |
| **日　　期：2023年12月24日** |

**电 子 科 技 大 学**

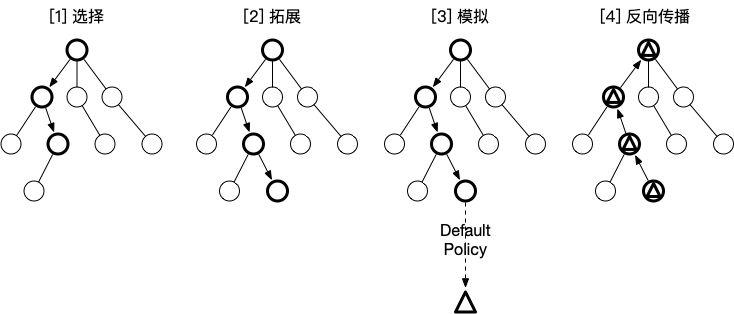
**实 验 报 告**

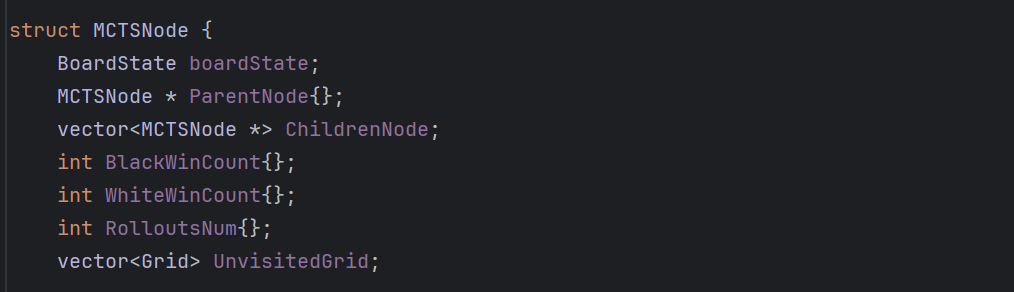
## 一、实验项目名称

围棋AI

## 二、实验原理

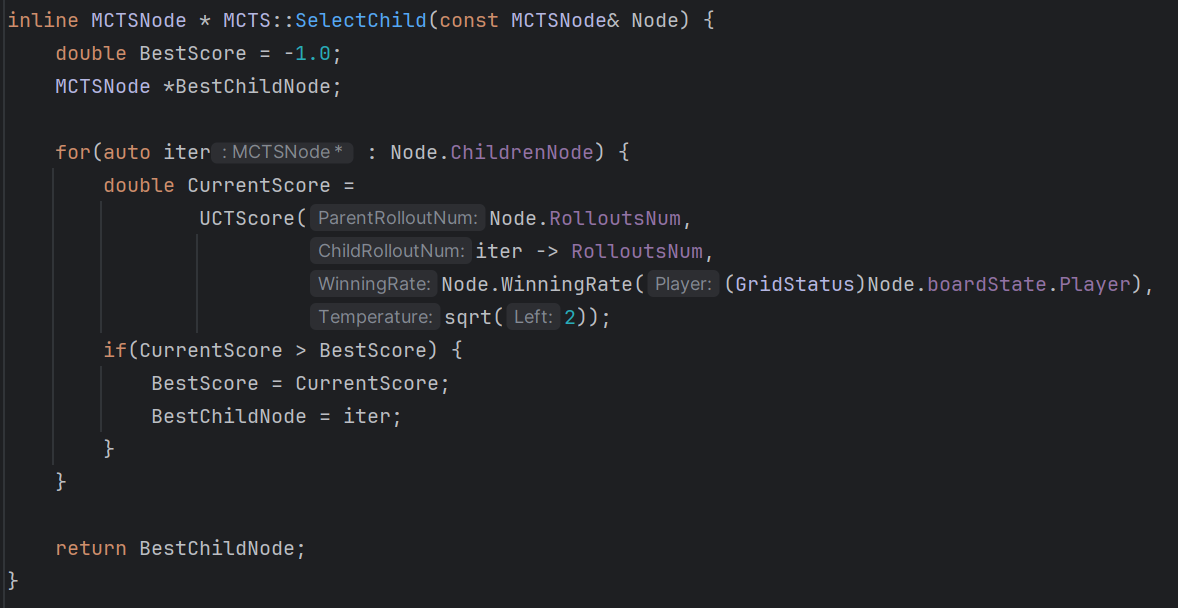
### 1、蒙特卡洛树搜索(Monte Carlo Tree Search，MCTS)：

 蒙特卡洛树搜索分为四步：选择(Selection)，拓展(Expansion)，模拟(Simulation)，反向传播(Back Propagation)。

蒙特卡洛树的节点包括七个基本信息：当前局面，父节点，子节点集合，黑棋获胜次数，白棋获胜次数和未扩展动作集合。

开始阶段，蒙特卡洛树只有一个根节点，即需要决策的局面，然后从根节点按四步开始进行搜索。

[1] 选择(Selection)

 在选择阶段，需要从根节点，即需要决策的局面R，向下选择一个价值最高的子节点N扩展，局面R是每次迭代中首个检查节点。

对于被检查的节点，有三种可能的情况：

1. 该节点所有可行动作都被扩展过
2. 该节点有可行动作还未被拓展过
3. 该节点的游戏状态为结束

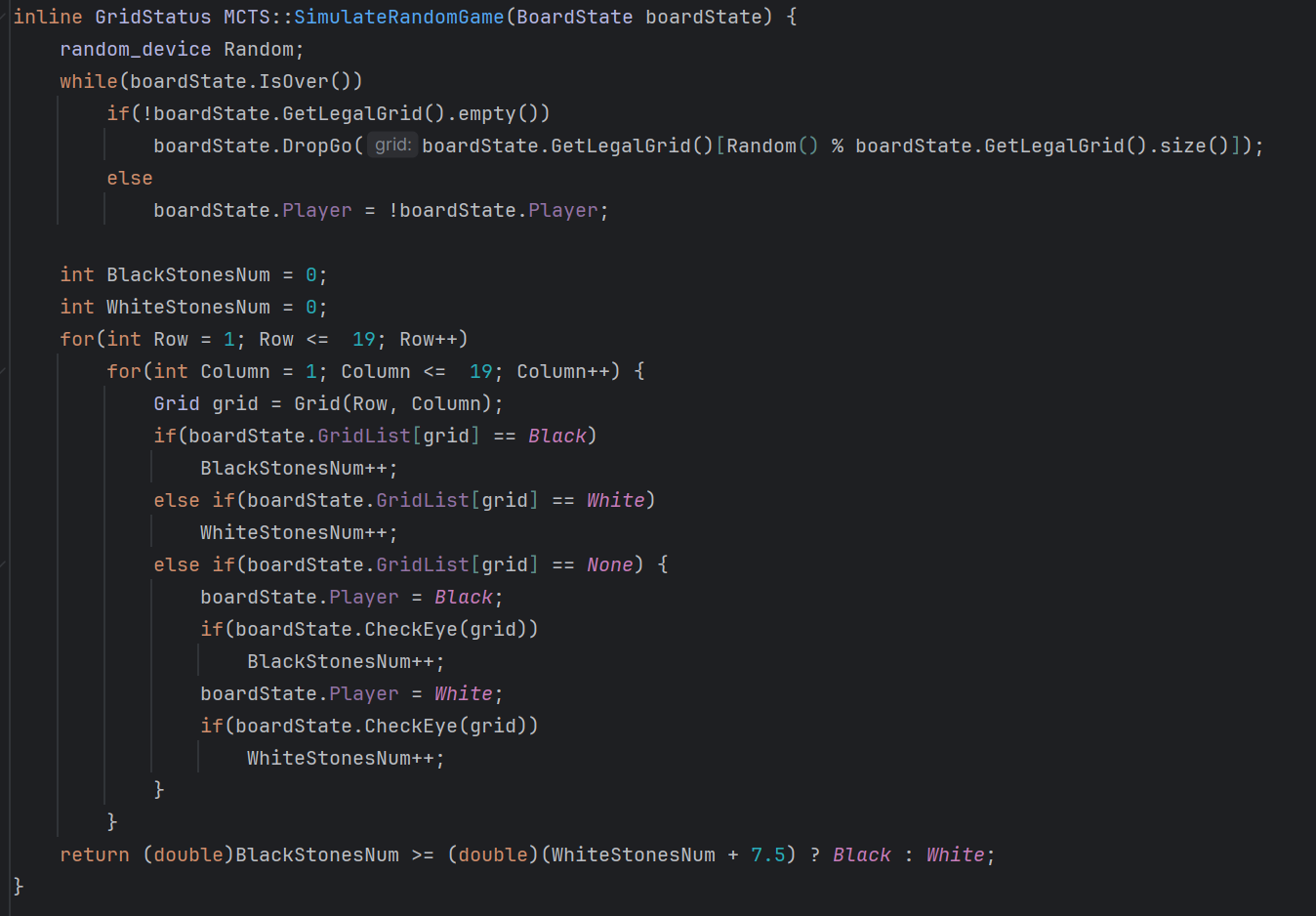
对于这三种可能：

1. 如果所有可行动作都被拓展过，使用UCT公式计算该节点所有子节点的UCT值，并找到UCT值最大的子节点继续检查。反复向下迭代
2. 如果被检查的局面存在没有被拓展的子节点，将此节点就是本次迭代的目标节点N，并找出N还未被拓展的动作A。执行步骤[2]
3. 如果检查的节点游戏状态为结束，执行步骤[4]

[2] 拓展(Expansion)

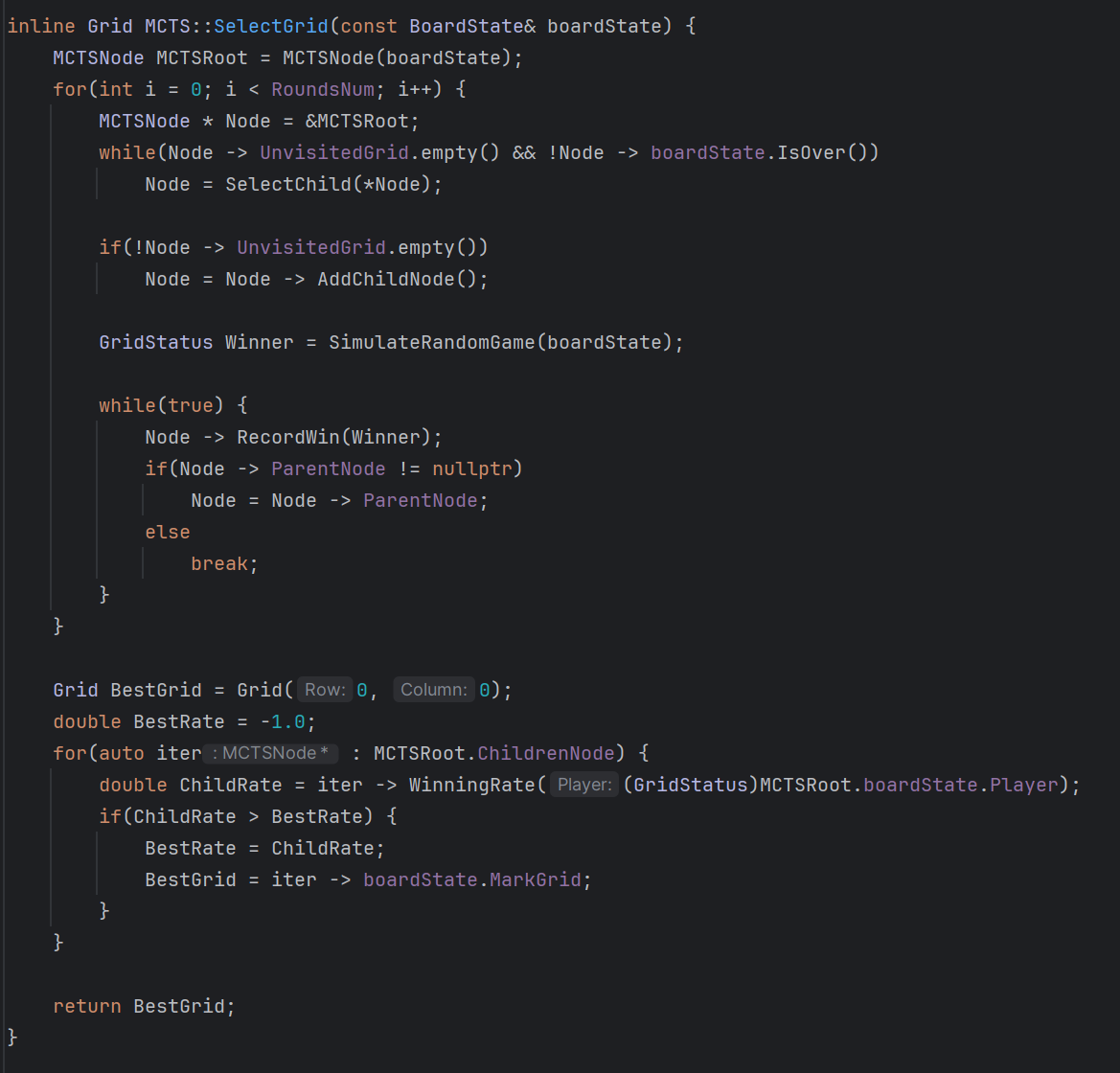
选择阶段结束后，将查找到价值最高的节点N，选择节点N的一个未拓展的动作A，在搜索树中创建一个节点M作为N的子节点。M的局面就是节点N在执行动作A后返回的局面。

[3] 模拟(Simulation)

从节点M开始，用快速走子策略（Rollout policy）将棋局进行到结束，得到一个胜负结果。将胜负结果作为评分，若胜利则评分为1，失败则评分为0

[4] 反向传播(Back Propagation)

在节点M的模拟结束后，根据模拟结果来更新M节点的所有父节点的累计评分。如果在[1]的选择中找到一个游戏状态为结束的节点，就根据该结果更新评分。

每一次迭代都会拓展搜索树，随着迭代次数的增加，搜索树的规模也不断增加。当到了一定的迭代次数或者时间之后结束，选择根节点下最好的子节点作为本次决策的结果。

### 2、信任度上限树(Upper Confidence bound applied to Trees，UCT)

在蒙特卡洛树中，每次迭代都选择胜率最大的节点进行模拟，这样会导致新节点无法被搜索到。为了在最大胜率和新节点搜索上保持平衡，UCT(Upper Confidence Bound，上限置信区间算法)被引入

UCT（Upper Confidence Bound，上限置信区间算法）公式：

其中是 节点的胜利次数，是节点的模拟次数，是总模拟次数， 是探索常数，理论值为 ，可根据经验调整。

UCT算法使用了置信区间的上限值作为局面价值，根据公式，节点的模拟次数越少，价值会越高。依据蒙特卡洛树搜索算法，价值越高的点，被选择的概率越高。使被探索比较少的点，可以获得更多的探索机会。

### 3、前馈神经网络(Feedforward Neural Network，FNN)

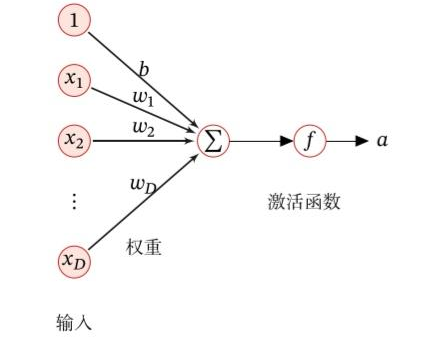
#### [1] 神经元(Artificial Neuron)

人工神经元(Artificial Neuron)，简称神经元(Neuron)，是构成神经网络的基本单元，其主要是模拟生物神经元的结构和特性，接收一组输入信号并产生输出。

假设一个神经元接受输入，令向量 来表示这组输入，并用净输入(Net Input) 来表示一个神经元所获得的输入信号x的加权和：

其中是一个维权重向量，是偏置。

净输入z在经过一个非线性函数后，得到神经元的活性值(Activation) ：

其中非线性函数 为激活函数(Activation Function)。

#### [2] 激活函数(Activation Function)

激活函数(Activation Function)是一种添加到人工神经网络中的函数，旨在帮助网络学习数据中的复杂模式。

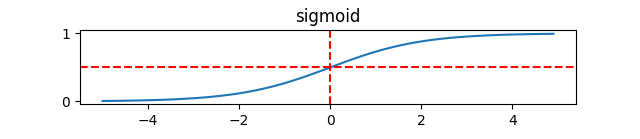
为了增强网络的表示能力和学习能力，激活函数需要具备以下几点性质：

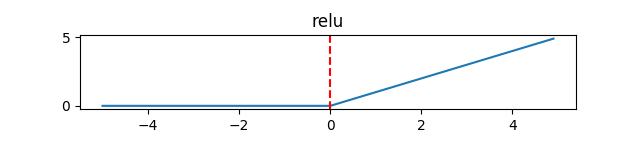
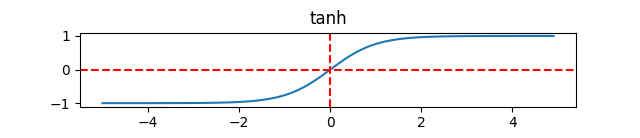
(1) 连续并可导(允许少数点上不可导)的非线性函数。可导的激活函数可以直接利用数值优化的方法来学习网络参数。

(2) 激活函数及其导函数要尽可能的简单，有利于提高网络计算效率。

(3) 激活函数的导函数的值域要在一个合适的区间内，不能太大也不能太小，否则会影响训练的效率和稳定性。

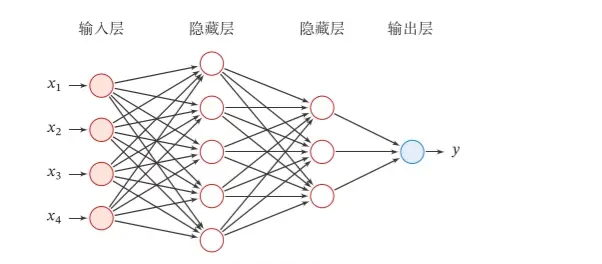
#### [3] 常见激活函数

1. Sigmoid函数：

1. Tanh函数
2. ReLu函数

#### [4] 前馈神经网络(Feedforward Neural Network，FNN)

前馈网络可以看作一个函数，通过简单非线性函数的多次复合，实现输入空间到输出空间的复杂映射。在前馈神经网络中，各神经元分别属于不同的层。每一层的神经元可以接收前一层神经元的信号，并产生信号输出到下一层。第0层称为输入层，最后一层称为输出层，其他中间层称为隐藏层．整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示。



|  |  |
| --- | --- |
| 符号 | 含义 |
|  | 神经网络的层数 |
|  | 第层神经元的个数 |
|  | 第层神经元的激活函数 |
|  | 第层到第层的权重矩阵 |
|  | 第层到第层的偏置 |
|  | 第层神经元的净输入 |
|  | 第层神经元的输出(活性值) |

前馈神经网络前向传播公式：

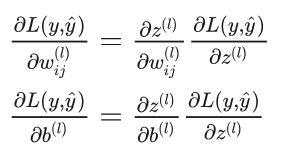
对每一层，递归地采用这个定义，预测结果可以表示为

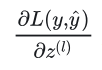
其中，可以视作每一层函数的复合函数

#### [5] 反向传播算法(Backpropagation)

假设采用随机梯度下降进行神经网络参数学习，给定一个样本，将输入到神经网络模型中，得到网络输出为

根据链式法则：

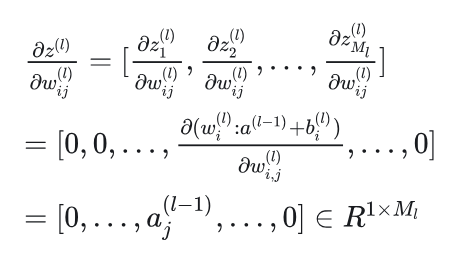


 其中， 是目标函数关于第层的神经元 的偏导数。



1. 计算

由于

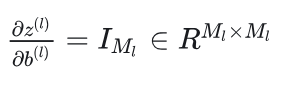
偏导数：



1. 计算

由于

偏导数：

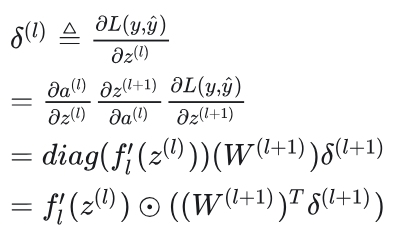




1. 计算

偏导数 表示第层神经元对最终损失的影响，称为第层神经元的误差项，用 表示。

偏导数：



其中 ⊙ 为哈达玛积符号，表示两个向量逐个元素相乘

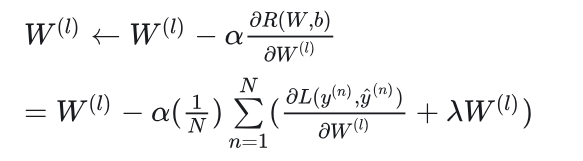
根据上述运算，参数梯度：

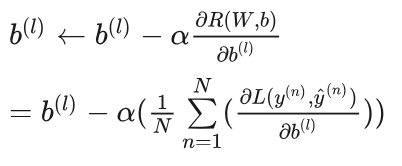






参数更新:



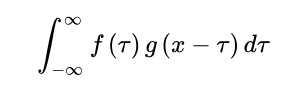


其中为学习率，为一批次样本大小。

### 4、卷积神经网络(Convolutional Neural Networks，CNN)

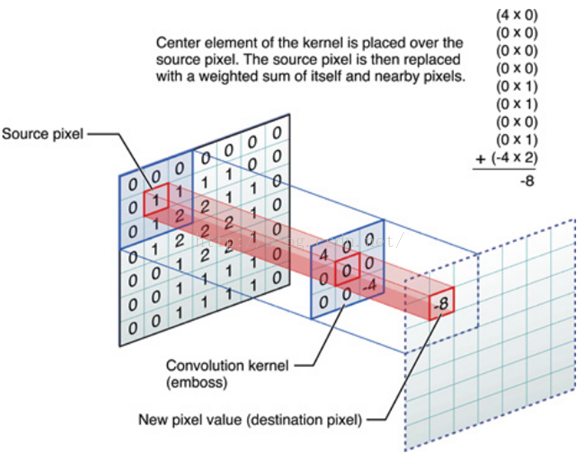
#### [1] 卷积(Convolution)

设是上的两个可积函数，作[积分](https://baike.baidu.com/item/%E7%A7%AF%E5%88%86/5749068?fromModule=lemma_inlink)：



可以证明，关于几乎所有的实数x，上述积分是存在的。随着x的不同取值，这个积分就定义了一个新函数 ，称为函数与的卷积，记为 。

物理意义为：系统某一时刻的输出是由多个输入共同作用(叠加)的结果。

在卷积神经网络中，为输入数据点，输入数据矩阵为所有输入数据点的叠加。

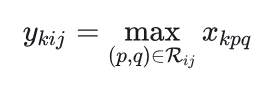
称为作用点，卷积核为所有作用点的叠加。

#### [2] 池化(Pooling)

池化(Pooling)的本质是对数据进行下采样。池化(Pooling)对于输入的特征图(Feature Map),使用某种方式对其进行降维压缩，加快运算速度。

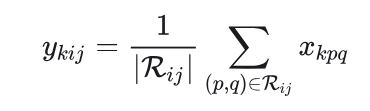
主要的池化方式有一下两种：

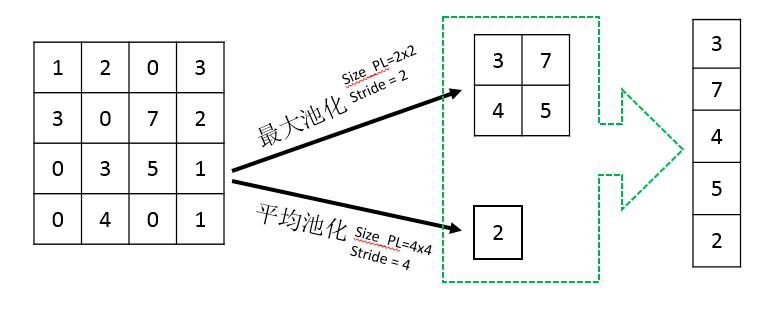
(1) 最大池化(Max Pooling) :

最大池化(Max Pooling)将输入的图像划分为若干个矩形区域，对每个子区域输出最大值。

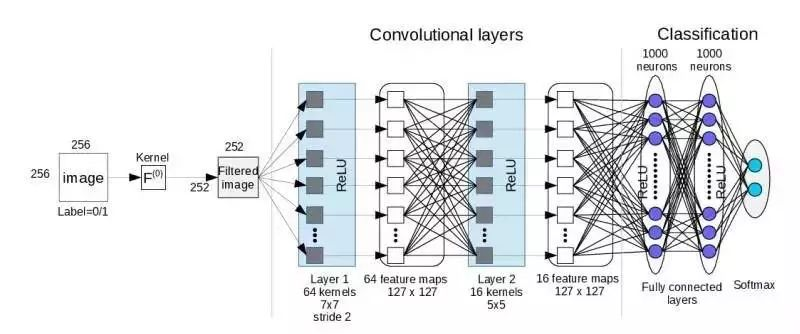
其中， 表示与第个特征图有关的在矩形区域的最大池化输出值，表示矩形区域中位于的元素。

(2) 平均池化(Average Pooling):

平均池化(Average Pooling)是将输入的图像划分为若干个矩形区域，对每个子区域输出所有元素的平均值。

其中， 表示与第个特征图有关的在矩形区域的平均池化输出值，表示矩形区域中位于的元素。

#### [3] 卷积神经网络(Convolutional Neural Networks，CNN)

卷积神经网络(Convolutional Neural Networks)是一种深度学习模型或类似于人工神经网络的多层感知器。

卷积神经网络(Convolutional Neural Networks)主要由以下5层组成：

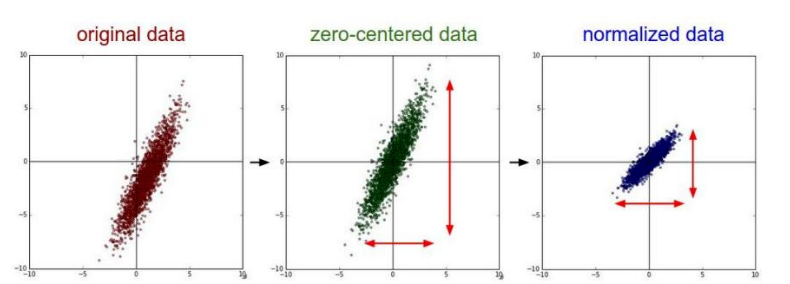
(1) 数据输入层：

数据输入层的作用主要是对原始数据进行预处理，其中包括：

去均值：将输入数据各个维度都中心化为0

归一化：将数据的取值范围归一化到同样的范围，减少

数据取值范围的差异的干扰。



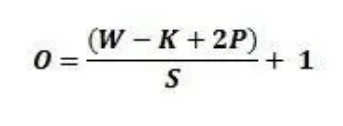
(2) 卷积层：

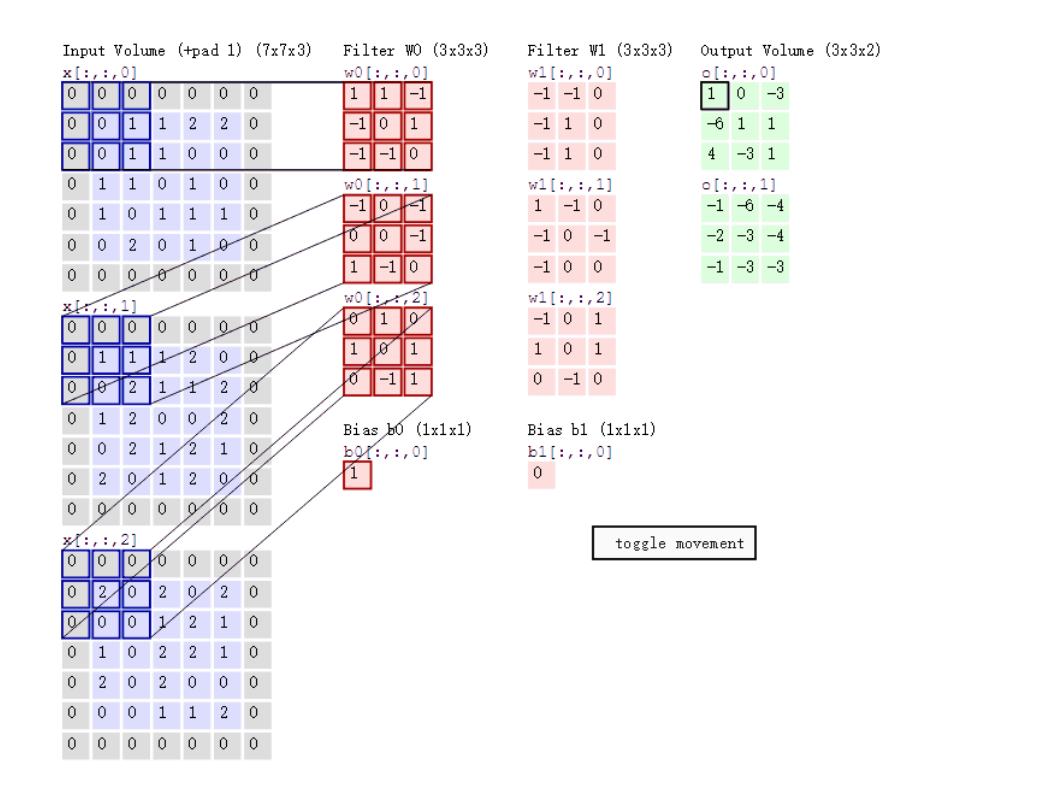
卷积层的作用主要是对原始数据进行卷积操作，将输入数据中符合卷积核的特征提取出来。即用卷积核在输入数据上滑动，进行卷积运算，得到特征图(feature map)。

卷积操作中主要包括以下两个参数：

·步幅(stride)：过滤器一次移动的距离。

* 零填充 (zero-padding)：在卷积运算前，在输入数据矩阵外围补充0的圈数。

 卷积层输出大小公式为：

 其中，为输出尺寸，为滤波器尺寸，为零填充，为步幅。

(3) ReLu激励层：

卷积神经网络采用的激活函数一般为ReLu(The Rectified Linear Unit，修正线性单元)，它的特点为收敛快，求梯度简单。

有时也会使用 Leaky ReLu 或 Maxout。

(4) 池化层：

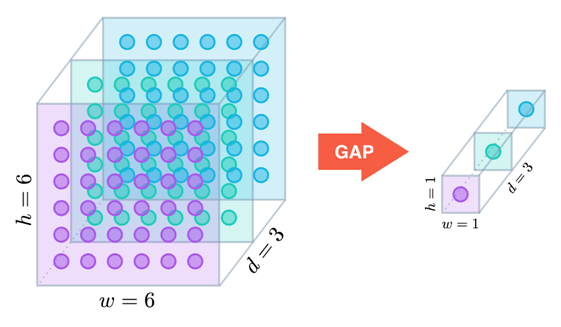
池化层位于卷积层之间，主要作用是对非线性激活后的结果进行降采样，以减少参数的数量，避免过拟合，并提高模型的处理速度。

具体作用：

·压缩数据：缩小输入数据矩阵的大小，并保证特征的尺

度不变。

·特征降维：将冗余信息剔除，提取重要特征。

·防止过拟合

(5) 全连接层

全连接层位于卷积神经网络尾部，两层之间所有神经元都有权重连接，与传统神经网络神经元的联系方式相同。

### 5、归一化处理(normalization)

归一化的目的就是使得预处理的数据被限定在一定的范围内，从而消除奇异样本数据导致的不良影响。

奇异样本数据是指相对于其他输入样本特别大或特别小的样本矢量（即特征向量）。



奇异样本数据的存在会引起训练时间增大，同时也可能导致无法收敛，因此，当存在奇异样本数据时，在进行训练之前需要对预处理数据进行归一化；反之，不存在奇异样本数据时，则可以不进行归一化。

常见归一化方法：

1. 最大最小标准化(Min-Max Normalization):

(1) 该方法将原始数据线性化的方法转换到[0,1]的范围。

(2) 该方法适用在数值比较集中的情况。

(3) 缺陷：若max和min不稳定，容易使得归一化结果不

稳定。

应用场景：在不涉及距离度量、协方差计算、数据不符合

正太分布的时候，可以使用该方法，比如图像

处理中，将RGB图像转换为灰度图像后将其

值限定在[0, 255]的范围。

(2) Z-score标准化(Z-score Normalization):

其中，、分别为原始数据集的均值和方差。

1. 该方法将原始数据集归一化为均值为0、方差1的数

据集。

1. 该种归一化方式要求原始数据的分布可以近似为高斯分布

应用场景：在分类、聚类算法中，需要使用距离来度量相

似性的时候、或者使用PCA技术的时候，Z-

score标准化表现更好。

(3) 神经网络归一化:

该方法常用在数据分化较大的场景，通过数学函数，将

原始值进行映射。

该方法包括 log、正切等，需要根据数据分布的情况，

决定非线性函数的曲线：

1. log对数函数归一化：

其中max表示样本数据的最大值，并且所有样本

数据均要大于等于1。

1. 反正切函数归一化：

该方法将大于等于0的数据映射到[0, 1]区间上，

将小于0的数据映射到[-1, 0]上。

(4) 批标准化(Batch Normalization):

通常来说，数据标准化预处理对于浅层模型就足够有效：处理后的任意一个特征在数据集中所有样本上的均值为0、标准差为1。

但对深层[神经网络](https://so.csdn.net/so/search?q=%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C&spm=1001.2101.3001.7020)来说，即使输入数据已做标准化，训

练中模型参数的更新依然很容易造成靠近输出层输出的剧烈变化。这种计算数值的不稳定性通常令我们难以训练出有效的深度模型。

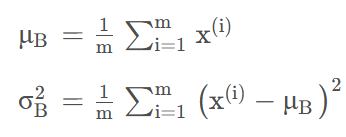
批标准化利用小批量上的均值和标准差，不断调整神 经网络中间输出，从而使整个神经网络在各层的中间输出的数值更稳定。

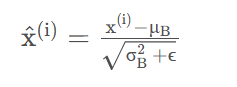
1. 全连接层：

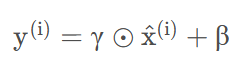
批标准化层置于全连接层中的仿射变换和激活函数之间。设全连接层的输入为，权重参数和偏差参数分别为和，激活函数为。设批标准化的运算符为。那么，使用批标准化的全连接层的输出为

其中，为全连接层的输出。

考虑一个由个样本组成的小批量，仿射变换的输出，

即批标准化层的输入。对于小批量中任意样本批标准化层的输出同样是维向量。  
 首先求小批量的均值和方差：

 再标准化全连接层的输出：

 然后引入两个可学习的模型参数，拉伸（scale）参数和偏移（shift）参数。这两个参数和形状相同，都为维向量。与分别做按元素乘法和加法计算：

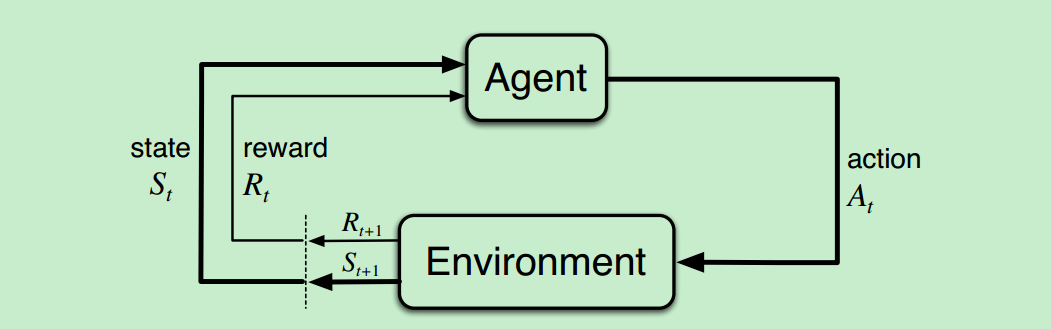
得到的批量归一化的输出。

1. 卷积层：

对卷积层来说，批量归一化发生在卷积计算之后、应用激活函数之前。如果卷积计算输出多个通道，对这些通道的输出分别做批标准化，且每个通道都拥有独立的拉伸和偏移参数，并均为标量。设小批量中有个样本。在单个通道上，假设卷积计算输出的高和宽分别为和。对该通道个元素同时做批标准化。对这些元素做标准化计算时，使用相同的均值和方差，即该通道中个元素的均值和方差。

### 6、强化学习(Reinforcement learning，RL)

强化学习(Reinforcement learning，RL)讨论的问题是一个智能体(agent)怎么在一个复杂不确定的环境(environment)里面去极大化它能获得的奖励。通过感知所处环境的状态(state)对动作(action)的 反应(reward)，来指导更好的动作，从而获得最大的收益(return)，这被称为在交互中学习，这样的学习方法就被称作强化学习。



## 三、实验过程

### 1、构建围棋游戏

在实现围棋AI前，首先构建围棋游戏部分，实现基本的围棋规则和人机对弈。

#### [1] 落子、提子

如果一颗棋子相邻4口气都被对方棋子占据，就会因为气尽而被吃掉。如果使用简单的二维数组跟踪棋盘中每一个交叉点的状态，想要判断是否有提子，在落下一颗棋子之后，就需要检查其所有相邻的对方棋子，检查是否能吃掉他们。具体来说，需要检查以下几个内容：

·检查相邻的棋子是否只剩一口气

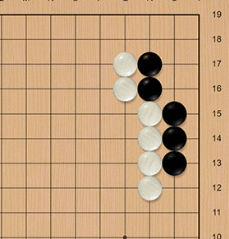
·接着检查这些邻居的相邻棋子是否还有气剩余

·继续检查这些邻居的邻居的相邻棋子，如此循环。

这个过程可能需要数百个步骤才能完成，降低了检查效率。

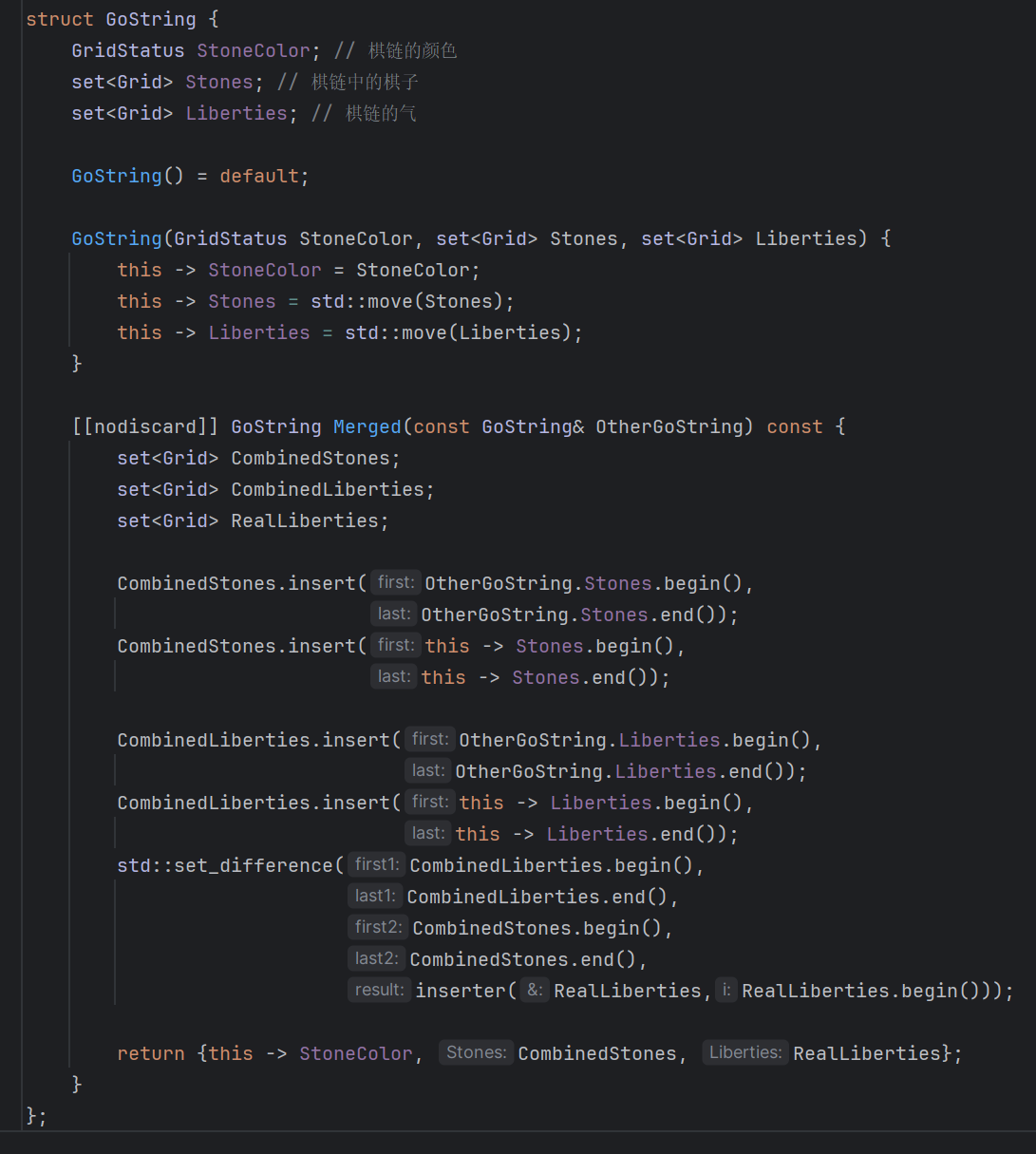
因为独立检查各个棋子会增加计算的复杂度，因此将相邻的同色棋子组合为一个整体，同时跟踪这个整体的状态和它们的气，这样做可以在实现棋盘逻辑时获得更高的效率。

将一组同色相邻的棋子称为一条棋链。如下图所示。



其中，用四种颜色标出了四条不同的棋链。

棋链的实现如下：

其中，Merged函数实现了将两个棋链合并，并更新棋链的气和棋子。

落子动作的算法如下所示：

(1) 合并任何同色且相邻的棋链

(2) 减少对方所有相邻棋链的气

(3) 如果对方某条棋链气尽了，则提走它们

此外，如果新生成的棋链气数为0，则拒绝这个动作。

#### [2] 劫争

如果一次落子动作导致棋盘完全返回上一个出现过的状态，就会触发劫争规则。

因此，在代码中编写如下规则：棋手一次落子动作不能让棋局

恢复到上一个回合的游戏状态，游戏状态包括棋盘上所有棋子的位置，以及下一回合的执子方。这种规则称为局势大劫规则。

#### [3] 终盘

在终局后需要判断双方的胜负，通常在终局后每块棋的死活都

已经确定，因此可以使用以下几个启发式规则帮助机器人判断棋

局的胜负：

·不要在完全被同色棋子所包围的区域落子

·不要选择落子后会导致只剩一口气的操作

·如果对方只剩一口气，总是吃掉它

根据这些启发式规则，可以让机器人继续已经终局的棋局，直

到尽量填满所有可以填满的空点，再利用中国计数法对棋局的胜

负进行判断。

### 2、使用树搜索下棋

树搜索算法是一类搜索策略，它们循环遍历许多可能的决策序

列，并找到可能产生最佳结果的决策，比如极小化极大搜索算法和

蒙特卡洛树搜索。

由于在围棋游戏中，对棋局进行评估是非常困难的事：遵循基于

提子数量的简单启发式规则无法胜过大多数棋手。因此，极小化极

大搜索算法无法应用在围棋游戏中。

蒙特卡洛树提供了一种方法，可以在不依赖任何围棋策略知识

的前提下评估游戏状态。蒙特卡洛树不需要利用游戏的启发式规则，

而是通过模拟随机棋局来评估棋局的好坏。

蒙特卡洛树搜索的每一轮计算都包括3个步骤：

(1) 将新的棋局添加到蒙特卡洛树中

(2) 从这个棋局开始模拟随机对弈

(3) 根据随机对弈的结果更新树节点的统计数据

在允许的时间内，多次重复此过程，最后搜索树的根节点的统计

数据会给出最佳动作。

但在每一回合中游戏AI能使用的时间是有限的，这意味着能够

执行的推演次数也是有限的。每多一次推演，都能提高一个可能动

作的评估水平。将推演看作某种总量有限的资源，如果在动作A上

多消耗一次推演，在动作B上就少消耗一次，因此需要一个策略来

决定如何分配有限的预算。

标准的策略称为信任度上限树(Upper Confidence bound

applied to Trees，UCT)，其可以在下面两个相互冲突的目标之间

取得一定平衡。

第一个目标是花费更多时间去检查最佳的动作。将这个目标称为

深入挖掘，即对至今为止搜索到的理想目标进行深入挖掘。另一个

目标为花费更多时间来提高被访问次数较少的分支的评估率，这个

目标称为广泛搜索。

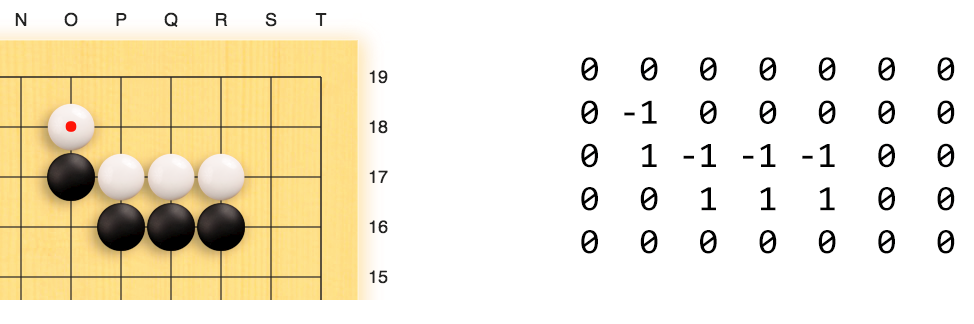
### 3、在围棋游戏中使用神经网络

虽然蒙特卡洛树可以对游戏状态进行评估，但由于其利用随机性来分析复杂的局面的特性，往往需要进行大量棋局的模拟，导致花费时间过长且得到的结果准确性较低。

为解决这两点问题，在围棋游戏中引入神经网络来预测围棋游戏中任意给定棋局的下一步动作。

#### [1] 棋盘编码器

在构建神经网络前，需要对围棋游戏的状态进行编码，使游戏状态转换为输入数据矩阵，便于神经网络处理。

 首先考虑使用简单的二维数组进行编码，将本方棋子标记为1，对方棋子标记为 -1，空点标记为 0，称为单编码层。将本方和对方的棋子分别标记，而不是标记黑白棋子，这样将数据输入神经网络时，无需输入当前落子方，也便于神经网络进行统一的训练。

但如果将单编码层作为编码方式得到的数据作为神经网络的输入数据进行训练，会发现训练效果不佳，有两个原因：

·卷积操作是线性的，因此 +1 和 -1 在卷积操作后，会互相抵

消，使得我们很难单独分析某一方的棋形情况，必然会受到另

一方的棋子的影响。

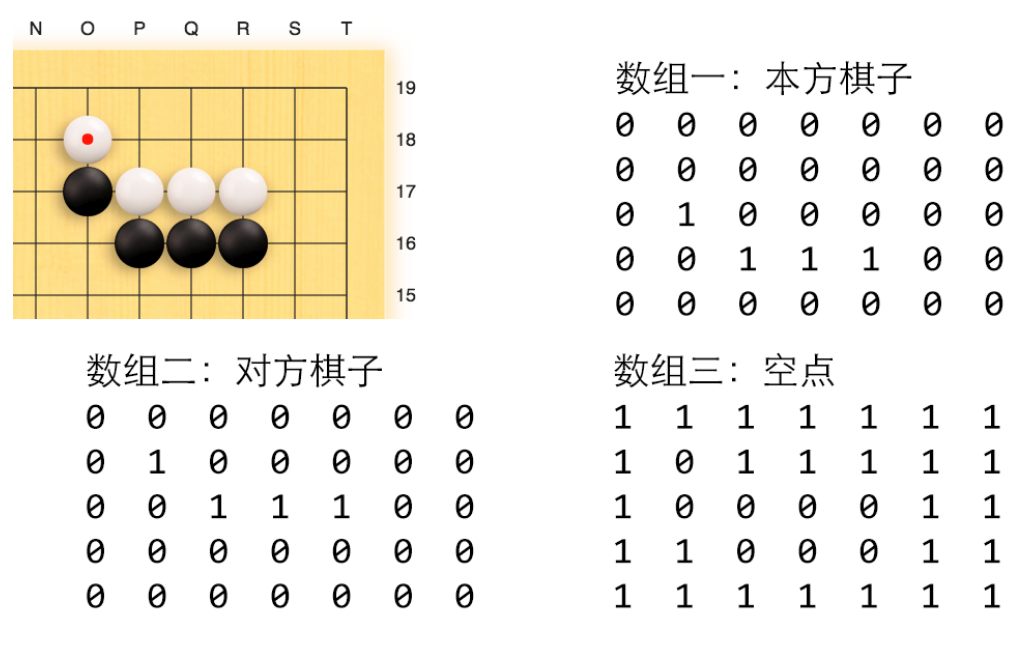
·忽略了围棋中棋子的气的概念

更适合作为神经网络的输入的棋盘编码方式，是one-hot 编码，即用 1 代表有某个性质的地方，用 0 代表没有某个性质的地方。

为了可以独立对一方的棋形进行分析，建立三个特征平面储存棋盘交叉点的状态：

·特征平面一：其中1代表本方棋子，0代表其他

·特征平面二：其中1代表对方棋子，0代表其他

·特征平面三：其中1代表空点，0代表其他

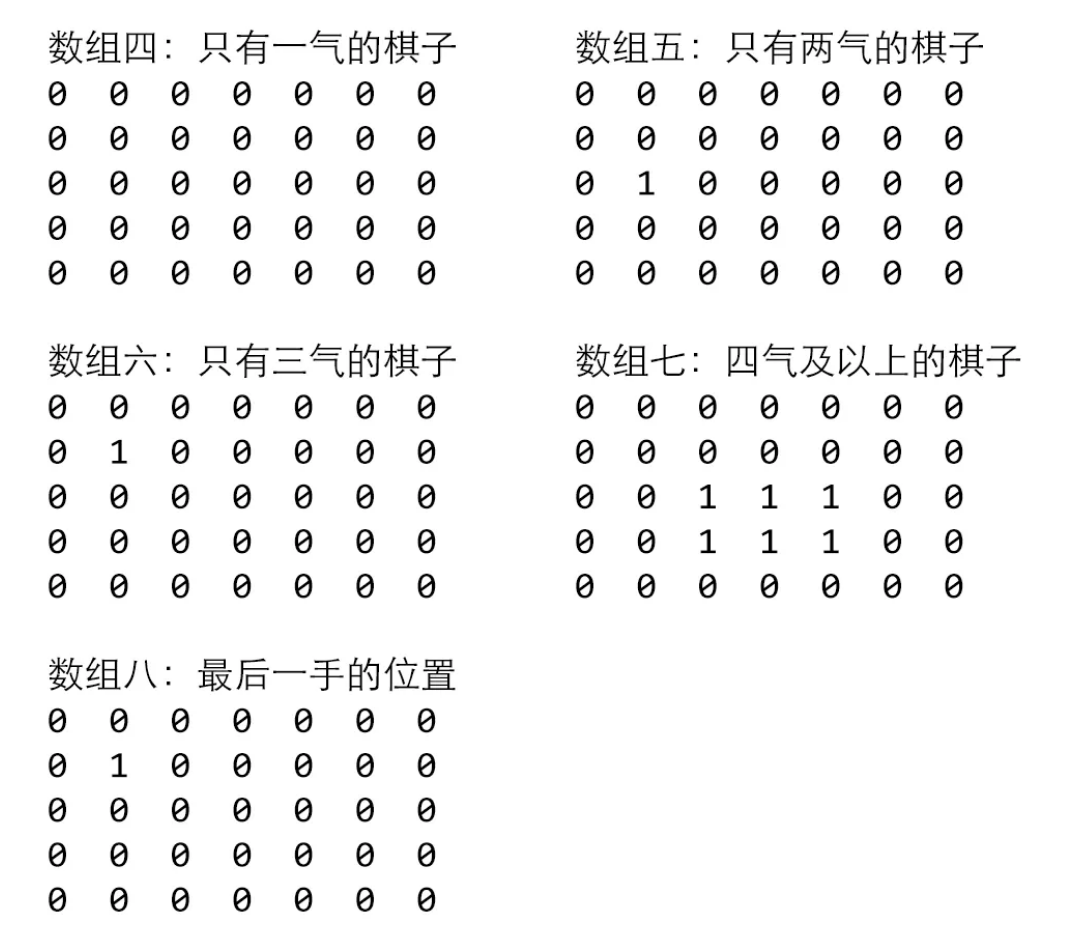
但这样编码依旧忽略了围棋中棋子的气的概念。为了引入棋子的气的概念，在原有的三个特征平面上再加入五个特征平面：

·特征平面四：其中1代表只有一气的棋子，0表示其他

·特征平面五：其中1代表只有两气的棋子，0表示其他

·特征平面六：其中1代表只有三气的棋子，0表示其他

·特征平面七：其中1代表四气及以上的棋子，0表示其他

·特征平面八：其中1代表最后一手的位置，0表示其他

其中，使用最后一手的位置，是希望尽可能拟合人类棋手的行为，回应对手的上一步。

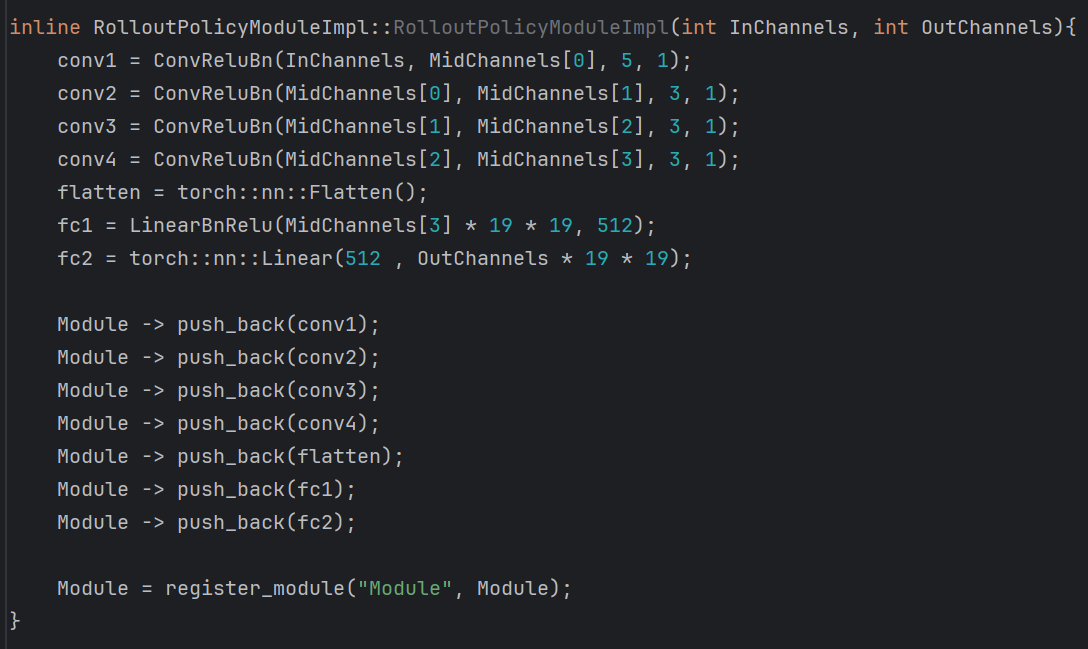
#### [2] 神经网络

构建好棋盘编码器之后，就需要为围棋数据设计神经网络。

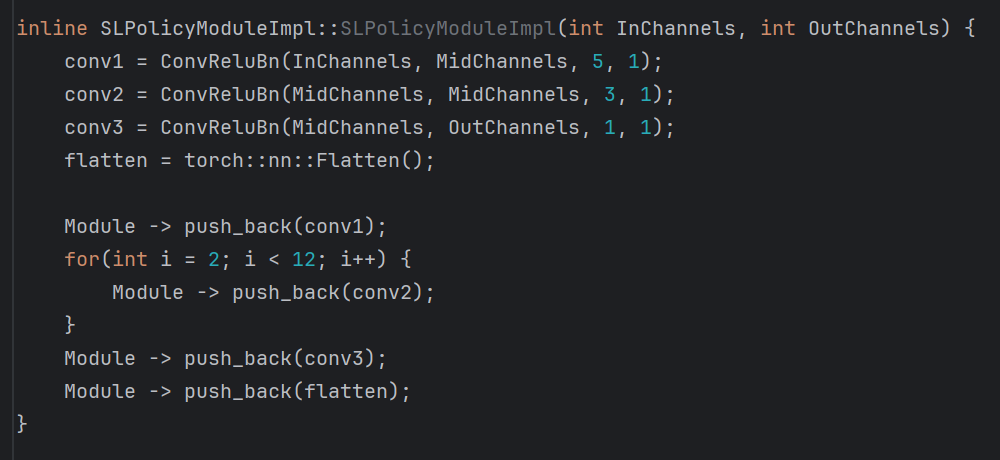
首先，考虑使用前馈神经网络(Feedforward Neural Network，FNN)

作为预测网络，但由于前馈神经网络的输入数据为向量，故在处理输入数据时需要将特征平面展开为向量，导致围棋棋形特征丢失，训练效果不佳。

为了保证围棋棋形特征不丢失，并利用这些特征获得对下一步的预测，因此引入卷积神经网络作为神经网络的主体。

 首先，构造一个小型的卷积神经网络，由四个使用批标准化的卷积层，一个展平层，一个使用批标准化的全连接层以及一个普通的全连接层：

使用这个神经网络训练，再小型数据集上可以获得一个较好的预测结果，准确率可以达到80%左右，但再大型数据集上依然无法达到所需的准确度，因此需要构建一个大型的卷积神经网络作为预测网络：

 上图为一个大型的卷积神经网络，其中包括了十三个使用批标准化的卷积层和一个展平层，可以做到对棋局较好的预测，在大型数据集上也可以达到较高的准确率。

#### [3] 强化学习

虽然卷积神经网络可以做到对人类高手的动作进行预测，但其很难做到超越人类高手，而且人类高手的行棋也不一定为最优下法，

为了超越人类高手，在神经网络中引入强化学习的概念。

强化学习的过程为：

·让神经网络进行自我对弈，并将结果保存为经验数据

·使用策略梯度方法，根据经验数据调整权重参数

其中，策略学习方法工作流程如下：

·当代理获胜时，增大它所选择过的动作的概率

·当代理失败时，减少它所选择过的动作的概率

具体实现方法为让神经网络自我对弈，并将每一局的胜负和每一步的棋局状态记录下来。在代理获胜的棋局中，则增大其选择的动作的概率，因此将代理选择的动作作为当前棋局对应的最佳动作，在目标向量对应的位置填入1。而在代理失败的棋局中，则需要减小所选动作的概率，使更新效果和获胜棋局相反，因此在目标向量对应的位置填入-1，这样做会改变损失函数梯度的正负值，使权重向相反的方向偏移。

将得到的经验数据用于训练网络，就实现了神经网络的强化学习。

### 4、使用神经网络改进蒙特卡洛树搜索

在构建完神经网络后，就可以利用这些网络改进蒙特卡洛树搜索。

第一种方法就是在推演中不在随机落子，而是使用一个策略网络

预测下一步的选点，而这个策略网络的速度需要足够快，才能在短

时间内完成大量推演，称为快策略网络。因此，选择构建的第一个

简单的神经网络作为快策略网络。

第二种方法是当到达搜索树的一个叶节点时，不再随机选择一个

新节点来扩展，而是向策略网络查询作。策略网络会给出所有候选

动作的概率分布，而每个节点都可以跟踪这个概率分布，因此根据

策略网络得到的更好的动作可以比其他动作更有机会选中。将这些

节点上的概率分布称为先验概率。而这个策略网络准确度要足够高，

称为强策略网络，因此将构建的第二个神经网络作为强策略网络。

第三种方法是对搜索树的一个叶节点进行模拟时，不模拟到棋局

结束，而是模拟到一定深度后，用价值网络对局面进行价值评估，

得到局面的价值来决定是否选择该动作。

根据这三种优化方法，可以构造出优化的蒙特卡洛树，步骤如下：

·在选择动作前，在游戏树上进行N次模拟

·每次模拟一直进行前瞻搜索，直到到达指定深度

·如果一个节点没有子节点，参考强策略网络提供的先验概率，

为每个合法的动作添加新的子节点，以扩展搜索树

·如果节点有子节点，则选择能够最大化价值的动作

·在棋盘上模拟本次选择的子节点

·当到达指定深度时，从价值函数获得价值函数，与策略推演的

输出值进行综合平均，计算这个节点的评估值

·根据模拟中涉及的叶节点更新所有节点

其中，最大化价值的动作由以下公式得到：

公式符号解释如下：

·表示选择使达到极大值的参数

·为游戏状态的访问次数

·为价值网络对节点的计算结果

·代表节点下的一次快策略推演的结果

·是一个0 – 1之间的数值，这里设为0.5

·为一个常数，默认设为5倍

·为当前节点的父节点的访问次数

·为策略网络对动作给出的预测概率

## 四、实验成果

实现了围棋游戏，并设计和实现了围棋AI的基本逻辑，经过训练，实现了可对弈的围棋AI